TUGAS AKHIR – SS 141501

KLASIFIKASI ENZIM PADA *DATABASE DUD-E* DENGAN METODE *LOGISTIC REGRESSION ENSEMBLE* (*LORENS*)

JAINAP NIKEN MELASASI
NRP 1311100 049

Dosen Pembimbing
Dr.rer. pol. Heri Kuswanto, S.Si., M.Si.

Program Studi S1 Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2015

TUGAS AKHIR – SS 141501

KLASIFIKASI ENZIM PADA *DATABASE DUD-E* DENGAN METODE *LOGISTIC REGRESSION ENSEMBLE* (*LORENS*)

JAINAP NIKEN MELASASI
NRP 1311100 049

Dosen Pembimbing
Dr.rer. pol. Heri Kuswanto, S.Si., M.Si.

Program Studi S1 Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2015

FINAL PROJECT – SS 141501

ANALYSIS OF RISK FACTORS AFFECTING THE NUMBER OF MALARIA CASES IN EAST JAVA DURING 2013 USING GEOGRAPHICALLY WEIGHTED NEGATIVE BINOMIAL REGRESSION (GWNBR)

NURINA HAYU RATRI
NRP 1311 100 057

Supervisor
Dr. Purhadi, M.Sc

Undergraduate Programme of Statistics
Faculty of Mathematics and Natural Sciences
Sepuluh Nopember Institute of Technology
Surabaya 2015

LEMBAR PENGESAHAN

KLASIFIKASI ENZIM PADA *DATABASE DUD-E* DENGAN METODE *LOGISTIC REGRESSION ENSEMBLE* (*LORENS*)

TUGAS AKHIR
Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada
Program Studi S-1 Jurusan Statistika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember

Oleh :
JAINAP NIKEN MELASASI
NRP 1311 100 049

Disetujui oleh Pembimbing Tugas Akhir

Dr.rer. pol. Heri Kuswanto, S.Si., M.Si.
NIP : 19820326 200312 1 004 (.....)

Mengetahui
Ketua Jurusan Statistika FMIPA-ITS

Dr. Muhammad Mashuri, MT
NIP. 19620408 198701 1 001

SURABAYA, JULI 2015

KLASIFIKASI ENZIM PADA *DATABASE DUD-E* DENGAN METODE *LOGISTIC REGRESSION ENSEMBLE* (*LORENS*)

Nama Mahasiswa : Jainap Niken Melasasi
NRP : 1311 100 049
Jurusan : Statistika
Dosen Pembimbing : Dr. rer. pol. Heri Kuswanto, S.Si., M.Si

Abstrak

Proses penemuan obat merupakan proses yang kompleks, memerlukan waktu yang lama, dan biaya sangat mahal. Sampai ditemukan alternatif pembuatan obat terbaru yaitu metode in silico yang terbukti dapat mempersingkat biaya dan waktu. Penapisan in silico dalam kaitannya dengan penyakit dilakukan untuk menemukan inhibitor potensial. Penelitian ini bertujuan untuk menemukan calon inhibitor baru dalam pembuatan obat dengan target enzim pada Database DUD-E yang meliputi 3 jenis enzim yaitu aofb,cah2 dan hs90a. Pada pembuatan obat klasifikasi senyawa dilakukan dengan tahapan docking score. Tujuan dari penelitian ini adalah memprediksi hasil docking score menggunakan metode statistik yang sesuai untuk klasifikasi. Masing-masing enzim terdiri dari senyawa penyusun yang berbeda-beda akan diklasifikasikan kedalam jenis inhibitor baik (ligand) dan inhibitor buruk (decoy). Pada penenlitian ini tahapan docking score dilakukan dengan metode regresi logistik biner dan logistic regression ensemble (Lorens). Metode regresi logistik biner menghasilkan akurasi 90,4 persen untuk enzim aofb, 91,7 untuk enzim cah2 dan 94 persen untuk enzim has90 sedangkan metode logistic regression ensemble (Lorens) menghasilkan ketepatan klasifikasi 88,95 persen untuk enzim aofb, 92,1 untuk enzim cah2 dan 100 persen untuk enzim hs90a. Metode logistic regression ensemble (Lorens) lebih baik dalam mengklasifikasikan senyawa enzim karena mampu menghasilkan treshold optimal namun metode ini tidak memiliki model yang dapat diinterpretasikan.

Kata Kunci*— *Aofb, Cah2, Docking score, DUD-E, Hs90a, In silico, Logistic Regression Ensemble, Regresi Logistik biner

(Halaman ini sengaja dikosongkan)

CLASSIFICATION OF ENZIM IN DATABASE DUD-E USING LOGISTIC REGRESSION ENSEMBLE (LORENS)

Name of Student	: Jainap Niken Melasasi
NRP	: 1311 100 049
Departement	: Statistics
Supervisor	: Dr. rer. pol. Heri Kuswanto, S.Si., M.Si

Abstract

Innovation drugs is complexs process, it need long time and high cost. Innovation drugs alternative is in silico method there are more effective because can save cost and time. Screening in silico in connection with the disease do to find potential inhibitor. This study purposed to find potential new inhibitors in the manufacture of the drug to the target enzyme in the Database DUD-E that includes three types of enzymes are aofb, cah2 and hs90a. In the manufacture of the drug substance classification done by stages docking score. Stages docking score will be performed on three types of enzymes in the database DUD-E by using appropriate statistical methods for classification. Each enzyme is composed of a compound different constituent will be classified into either type of inhibitor (ligand) and bad inhibitor (decoy). At this research docking score stage performed by binary logistic regression and logistic regression ensemble (Lorens). Classification accuracy of aofb enzim with binary logistic regression is 90,4 percent, 91,7 percent for cah2 enzim, and 94 percent for has90a enzim while classification accuracy of aofb enzim with logistic regression ensemble (Lorens) is 88,95 percent, 92,1 percent for cah2 enzim and 100 percent for hs90a enzim. Logistic regression ensemble (Lorens) method better in classifying the enzyme compounds because use optimal threshold.

Keywords— Aofb, Binary Logistic Regression, Cah2, Docking score, DUD-E, Hs90a, In silico, Logistic Regression Ensemble

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kepada Allah SWT, yang telah memberikan rahmat sehingga penyusunan Tugas Akhir ini dapat terselesaikan tepat waktu. Tugas Akhir yang berjudul "***Klasifikasi Enzim Pada Database DUD-E Dengan Metode Logistic Regression Ensemble (Lorens)***" ini disusun untuk memenuhi salah satu syarat kelulusan Program Studi S1 Jurusan Statistika FMIPA ITS.

Dengan terselesaikannya penyusunan Tugas Akhir ini, penulis mengucapkan terima kasih kepada:

1. Allah SWT yang telah memberikan kemudahan dan kelancaran dalam menjalankan Tugas Akhir sampai dengan penyusunan laporan.
2. Bapak, Ibu, Kakak dan Adik sebagai keluarga yang senantiasa memberikan dukungan baik moril maupun materil dan juga doa yang tiada henti.
3. Dr. rer. pol. Heri Kuswanto, S.Si., M.Si. selaku dosen pembimbing yang selalu memberikan pengarahan kepada penulis selama penyusunan laporan Tugas Akhir.
4. Dr. Brodjol Sutijo Suprih Ulama, M.Si. dan Dr. Irhamah., S.Si., M.Si selaku dosen penguji yang senantiasa memberikan kritik dan saran demi kesempurnaan Tugas Akhir ini.
5. Dr. Muhammad Mashuri, MT selaku Ketua Jurusan Statistika ITS.
6. Dra. Lucia Aridinanti, MS selaku Kaprodi S1 Jurusan Statistika ITS.
7. Dr. Dra. Kartika Fithriasari, M.Si. selaku dosen wali yang telah memberikan pengarahan selama proses perkuliahan.
8. Seluruh dosen Statistika ITS dan dosen non Statistika ITS yang telah memberikan ilmu-ilmu yang tiada ternilai harganya dan segenap karyawan jurusan Statistika ITS.
9. Rajif Sidik sebagai seseorang yang selalu memberikan dorongan, bantuan dan semangat dan perhatian selama ini.
10. Lucky, Marina dan Ita sebagai sahabat yang telah memberikan semangat dalam menyelesaikan Tugas Akhir ini.

11. Teman dan saudara seperjuangan Nurina, Onya dan Iin yang telah banyak membantu dan memberikan semangat serta bantuan.
12. Ayu Asfihani dan Mbak hani yang telah memberikan banyak pemahaman mengenai topik Tugas Akhir ini.
13. Seluruh teman-teman mahasiswa Statistika ITS khususnya S1 angkatan 2011 yang selalu memberikan semangat dan dorongan hingga terselesaikannya Tugas Akhir ini.
14. Semua pihak yang telah membantu dalam penulisan Tugas Akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari sepenuhnya bahwa laporan Tugas Akhir ini masih jauh dari sempurna. Oleh karena itu, penulis menerima kritik dan saran yang membangun bagi perbaikan di masa yang akan datang. Semoga laporan ini bermanfaat bagi penelitian selanjutnya.

Surabaya, Juli 2015

Penulis

DAFTAR ISI

(*Halaman ini sengaja dikosongkan*)

DAFTAR TABEL

(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Proses penemuan obat merupakan proses yang kompleks, memerlukan waktu yang lama, dan biaya sangat mahal. Pada umumnya, waktu untuk mengembangkan kandidat obat memerlukan waktu 5 (lima) tahun, saat memasuki fase uji klinis sampai menjadi obat komersial memerlukan waktu lebih dari 7 (tujuh) tahun, dengan total biaya lebih dari 700 juta dolar Amerika (DiMasi, Hansen, & Grabowski, 2002). Dewasa ini sebuah jalan alternatif dalam penemuan obat baru dimunculkan oleh para ahli computer dan teknologi informasi. Selama ini obat diuji coba sebelum di pasarkan dengan menggunkan metode *in vivio* dan *in vitro* saja namun sekarang muncul metode *in silico* atau di dalam computer. Penapisan *in silico* jauh lebih efektif dibandingkan dengan metode *in vivo* dan *in vitro*. Penapisan *in silico* dalam kaitannya dengan penyakit dilakukan untuk menemukan inhibitor potensial (Jenwitheesuk, 2008). Salah satu tahap yang paling penting dalam penemuan obat adalah identifikasi *lead compound*. Identifikasi *lead compound* merupakan tahap analisis struktur senyawa-senyawa *druggable* yang terpilih dan mengidentifikasi substruktur aktif umum, selanjutnya senyawa *novel* yang mengandung substruktur tersebut disintesis. Pada tahap ini pendekatan bioinformatika struktural dan kemoinformatika untuk menemukan obat sangat berguna (Markus & Edgar, 2004)

Penelitian ini bertujuan untuk menemukan calon inhibitor baru pada sebuah pembuatan desain obat dengan target enzim atau protein. Senyawa atau molekul penyusun enzim terdiri dari *ligand* atau disebut dengan inhibitor baik dan *decoy* atau disebut dengan inhibitor buruk. Suatu jenis enzim dengan kadar senyawa yang berbeda-beda dapat diklasifikasikan menjadi *ligand* atau disebut dengan inhibitor baik dan *decoy* atau disebut dengan inhibitor buruk. Pada teknik pembuatan obat klasifikasi senyawa enzim dilakukan dengan menggunakan *docking software* sebagai

alat utama untuk mensimulasikan pengikatan dari pencampuran (kandidat inhibitor baru) dengan target enzim atau protein berdasarkan struktur molekulnya. (Okada, Ohwada, & Aoki, 2013).

Proses *docking score* dilakukan terhadap 3 dari 102 jenis enzim pada database *DUD-E* dan menggabungkan perbandingan studi dengan 2 *docking software* (*Libdock dan CDocker*). *DUD-E* adalah standar *database* untuk simulasi *docking*. *Database* ini berisi 3 tipe data yaitu target enzim, *ligand* dan *decoys*. Tiga jenis target enzim yang akan diklasifikasikan berdasarkan kadar senyawa penyusunnya adalah *aofb*, *cah2* dan *hs90a*. *Docking* merupakan metode yang memprediksi orientasi yang disukai dari suatu molekul ketika *binding* dengan molekul lain untuk membentuk kompleks stabil (Lengauer, 1996). *Docking* paling sering digunakan untuk memprediksi orientasi *binding* kandidat obat molekul kecil dengan target proteinnya untuk memprediksi afinitas dan aktivitas molekul kecil. Tujuan dari penelitian ini adalah melakukan tahapan *docking score* dengan menggunakan metode statistik yang sesuai. Salah satu metode statistik yang dapat diterapkan untuk proses klasifikasi dengan dua kategori variabel respon adalah regresi logistik biner. Regresi logistik biner merupakan suatu metode analisis data yang digunakan untuk mencari hubungan antara variabel respon yang bersifat biner atau dikotomus dengan variabel prediktor yang bersifat polikotomus (Watson Hosmer & Lemeshow, 1995).

Penelitian menggunkan metode *logistic regression ensemble* (*Lorens*) sebelumnya pernah dilakukan oleh Lim, Ahn, Moon dan Chen pada tahun 2010 dengan studi kasus ekspresi gen pada ilmu kesehatana anak-anak untuk memprediksi *acute myeloid leukemia (AML)* dengan menggunakan 2 variabel respon yaitu *good prognosis complete remission (CR)* dan p*oor prognosis* atau *relapsed (R)* dan terdapat 53 pengamatan yaitu anak-anak berusia di bawah 15 tahun. Oleh karena itu akan diimplementasikan metode regresi logistik *ensemble* (*Lorens*) pada kasus klasifikasi enzim pada *database DUD-E*.

Pada penelitian oleh Lim, Ahn, Moon dan Chen pada tahun 2010 menyatakan bahwa klasifikasi dengan menggunakan regresi logistik untuk *high dimensional* data memerlukan pemilihan variabel ketika jumlah variabel prediktor terlalu banyak, kekurangan dari regresi logistik tersebut dapat disempurnakan oleh metode *logisitic regression ensemble* (*Lorens*) yang dikembangkan oleh Lim (2007) dengan menyertakan algoritma *Classification by Ensembles from Random Partition* (CERP). Pada metode *logisitic regression ensemble* (*Lorens*) beberapa model yang dihasilkan berasal dari partisi secara acak dari variabel prediktor yang digunakan. Model regresi logistik digunakan pada masing-masing sub ruang yang diperoleh dari partisi secara acak dari variabel prediktor.

Penelitian untuk klasifikasi enzim berdasarkan perhitungan *docking score* dengan menggunakan metode SVM telah dilakukan oleh Okada, Ohwada dan Aoki (2013). Metode SVM yang diperkenalkan oleh Vladimir Vapnik pada tahun 1995 memberikan hasil ketepatan klasifikasi 99% namun metode SVM memiliki kelemahan yaitu tidak memberikan keseimbangan antara *sensitifity* dan *spesitifity* dibandingkan dengan metode klasifikasi lainnya(Lim, Ahn, Moon dan Chen, 2010). Pada penelitian ini akan dilakukan klasifikasi dengan metode logistik regresi *ensemble* (*Lorens*) dengan harapan akan dihasilkan hasil klasifikasi yang lebih tinggi dibandingkan metode SVM. Metode SVM dan metode *logistic regression ensemble* (*Lorens*) memiliki kesamaan yaitu dapat mengklasifikasikan data linier dan bersifat *supervised*. Tujuan dari metode ini adalah memungkinkan melakukan prediksi dengan menggunakan model yang dihasilkan dari regresi logistik ensemble *high dimensional* data. Pengkasifikasian dilakukan dengan variabel prediktor yang telah dipartisi secara random(Lim, Ahn, Moon dan Chen, 2010). Dalam penelitian ini akan dikaji hasil klasifikasi senyawa enzim pada *database DUD-E* dengan menggunakan 2 pendekatan yaitu regresi logistik biner dan *logisitic regression ensemble* (*Lorens*).

1.2 Rumusan Masalah

Masalah dalam penelitian ini adalah memprediksi hasil klasifikasi senyawa enzim pada *Database DUD-E* kedalam jenis inhibitor baik (*Ligand*) dan inhibitor buruk (*decoy)*. Salah satu metode klasifikasi dengan respon biner adalah regresi logistik biner namun metode ini memiliki kelemahan dalam mengklasifikasikan data dengan respon positif dan negatif yang tidak seimbang. Kekurangan tersebut dapat disempurnakan oleh metode *logistic regression ensemble* (*Lorens*) dengan menggunakan pendekatan komputasioanl dimana metode *logistic regression ensemble* (*Lorens*) menggunakan *threshold* optimal untuk klasifikasi data. Disisi lain metode *logistic regression ensemble* (*Lorens*) tidak menghasilkan model yang dapat menjelaskan hubungan antar variabel prediktor dan respon. Sehingga kedua metode tersebut akan diterapkan dalam klasifikasi senyawa enzim pada *database DUD-E*. Regresi Logistik biner diterapkan untuk mendapatkan model yang dapat diinterpretasikan dan metode *logistic regression ensemble* (*Lorens*) diterapkan untuk mendapatkan ketepatan klasifikasi. Ketepatan klasifikasi dari kedua metode akan dibandingkan.

1.3 Tujuan

Penelitian ini dilakukan dengan tujuan mendapatkan model serta hasil klasifikasi menggunakan regresi logistik biner dan mendapatkan klasifikasi senyawa enzim pada *database DUD-E* menggunakan *logisitic regression ensemble* (*Lorens*). Untuk mengetahui performa dari masing-masing metode maka pada akhir pembahasan akan dilakukan perbandingan hasil klasifikasi menggunakan regresi logistik biner dan *logisitic regression ensemble* (*Lorens*).

1.4 Manfaat Penelitian

Manfaat dari penelitian ini bagi dunia medis diharapkan dapat memberikan informasi terhadap pembuatan jenis obat baru dengan kandungan senyawa pada target enzim yang diklasifikasikan ke dalam jenis inhibitor baik (*ligand*) dan

inhibitor buruk (*decoy*). Bagi Institusi pendidikan diharapkan dapat dijadikan sebagai bahan pengetahuan untuk penelitian selanjutnya yang lebih mendalam terkait dengan pembuatan jenis obat baru dengan kandungan senyawa pada target enzim yang diklasifikasikan ke dalam jenis inhibitor baik dan inhibitor buruk dengan menggunakan metode regresi logistik biner dan *logisitic regression ensemble* (*Lorens*).

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah dari 102 jenis enzim pada *database DUD-E* akan dilakukan *docking score* terhadap 3 jenis enzim yaitu *aofb, cah2* dan *hs90a* dengan jumlah partisi yang akan dicobakan pada masing-masing variabel sebesar 2,3,4,5,6,7,8,9 dan 10 serta jumlah *ensemble* sebanyak 10.

(Halaman ini sengaja dikosongkan)

BAB II
TINJAUAN PUSTAKA

2.1 Regresi Logistik

Regresi logistik merupakan salah satu metode *dichotomus* (berskala nominal atau ordinal dengan dua kategori) atau *polychotomous* (mempunyai skala nominal atau ordinal dengan lebih dari dua kategori) dengan satu atau lebih variabel prediktor. Untuk variabel respon pada regresi logistik bersifat kontinyu atau kategorik (Agresti, 1990).

2.2 Regresi Logistik Biner

Regresi Logistik Biner merupakan suatu metode analisis data yang digunakan untuk mencari hubungan antara variabel respon (y) yang bersifat biner atau dikotomus dengan variabel prediktor (x) yang bersifat polikotomus (Hosmer dan Lomeshow,1989). *Outcome* dari variabel respon y terdiri dari 2 kategori yaitu sukses dan gagal yang dinotasikan dengan y=1 (sukses) dan y=0 (gagal). Dalam keadaan demikian variabel y mengikuti distribusi Bernoulli untuk setiap observasi tunggal. Fungsi probabilitas untuk setiap observasi adalah diberikan sebagai berikut,

$$f(y) = \pi^{y}(1-\pi)^{(1-y)}; y = 0,1 \tag{2.1}$$

Dimana jika y=0 maka $f(y) = 1-\pi$ dan jika y=1 maka $f(y) = \pi$. Fungsi regresi logistik dapat dituliskan sebagai berikut

$$f(z) = \frac{1}{1+e^{-z}} \tag{2.2}$$

Dimana $z = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$

Nilai z antara $-\infty$ dan $+\infty$ sehingga nilai $f(z)$ terletak antara nilai 0 dan 1 untuk setiap nilai z yang diberikan. Hal ini

menunjukkan bahwa model logistik sebenarnya menggambarkan probablitias atau resiko dari suatu objek. Model Regresi logistik dituliskan sebagi berikut,

$$\pi(x) = \frac{e^{(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)}}{1 + e^{(\beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p)}} \qquad (2.3)$$

Dimana p=banyaknya variabel prediktor

Untuk mempermudah pendugaan parameter regresi maka model regresi logistik pada persamaan diatas dapat diuraikan dengan menggunakan transformasi logit dari $\pi(x)$ sehingga didapatkan persamaan

$$g_x = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \qquad (2.4)$$

Model tersebut merupakan fungsi linier dari parameter-parameternya. Dalam model regresi linier, diasumsikan bahwa amatan dari variabel respon diekpresikan sebagai $Y = E(Y|\mathrm{x}) + \varepsilon$ dimana

$$E(Y|\mathrm{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \qquad (2.5)$$

Merupakan rataan dari populasi dan ε merupakan komponen acak yang menunjukkan penyimpangan amatan dari rataan dan ε diasumsikan mengikuti sebaran normal dengan rataan 0 dan varians konstan.

Pada regresi logistik, variabel respon diekspresikan sebagai $y = \pi(x) + \varepsilon$ dimana ε mempunyai salah satu kemugkinan dari dua nilai yaitu $\varepsilon = 1 - \pi(x)$ dengan peluang $\pi(x)$ jka y=1 dan $\varepsilon = -\pi(x)$ dengan peluang $1 - \pi(x)$ jika y=0 dan mengikuti distribusi binomial dengan rataan nol dan varians $(\pi(x))(1 - \pi(x))$ (Hosmer dan Lomeshow,1989)

2.2.1 Estimasi Parameter

Estimasi parameter dalam regresi logistik dilakukan dengan metode maximum likelihood. Metode tersebut mengestimasi

parameter β dengan cara memaksimumkan fungsi likelihood dan mengasumsikan bahwa data harus mengikuti suatu distribusi tertentu. Pada regresi logistik, setiap pengamatan mengikuti distribusi Bernoulli sehingga dapat ditentukan fungsi likelihoodnya.

Jika x_i dan y_i adalah pasangan variabel bebas dan terikat pada pengamatan ke-i dan diasumsikan bahwa setiap pasangan pengamatan saling independen dengan pasangan pengamatan lainnya i=1,2, . . ., n maka fungsi probabilitas untuk setiap pasangan adalah sebagai berikut,

$$f\left(x_i\right)=\pi(x_i)^{y_i}\left(1-\pi\left(x_i\right)\right)^{1-y_i};y_i=0,1 \tag{2.6}$$

Dengan

$$\pi\left(x_j\right)=\frac{e^{\left(\sum_{j=0}^{p}\beta_j x_j\right)}}{\left(\sum_{j=0}^{p}\beta_j x_j\right)} \tag{2.7}$$

Dimana ketika j=0 maka nilai $x_{ij}=x_{i0}=1$

Setiap pasang pengamatan diasumsikan independen sehingga fungsi likelihoodnya merupakan gabungan dari fungsi distribusi masing-masing pasangan sebagai berikut,

$$l\left(\beta\right)=\prod_{i=1}^{n}f\left(x_i\right)=\prod_{i=1}^{n}\pi(x_i)^{y_i}(1-\pi(x_i)^{1-y_i}$$

$$=\left\{\prod_{i=1}^{n}(1-\pi\left(x_i\right))\right\}\left\{\prod_{i=1}^{n}e^{\left(log\left(\frac{g(x_i)}{(1-g(x_i))}\right)^{y_i}\right)}\right\}$$

$$=\left\{\prod_{i=1}^{n}\left(1-\pi(x_i)\right)\right\}e^{\left\{\sum_{i=1}^{n}y_i log\left(\frac{\pi(x_i}{(1-g(x_i))}\right)^{y_i}\right\}}$$

$$= \left\{ \prod_{i=1}^{n} \frac{1}{1+e^{\sum_{j=0}^{p}\beta_j x_{ij}}} \right\} e^{\left\{ \sum_{i=1}^{n} y_i \log\left(e^{\sum_{j=0}^{p}\beta_j x_{ij}} \right) \right\}}$$

$$= \left\{ \prod_{i=1}^{n} \left(1+e^{\sum_{j=0}^{p}\beta_j x_{ij}} \right)^{-1} \right\} e^{\left\{ \sum_{j=0}^{p} \left(\sum_{i=1}^{n} y_i x_{ij} \right) \beta_j \right\}}$$

Fungsi likelihood tersebut lebih mudah dimaksimumkan dalam bentuk log $l(\beta)$ dan dinyatakan dengan $L(\beta)$

$$L(\beta) = \log l(\beta) = \sum_{j=0}^{p} \left(\sum_{i=1}^{n} y_i x_{ij} \right) \beta_j - \sum_{i=1}^{n} \log \left(1+e^{\sum_{j=0}^{p}\beta_j x_{ij}} \right)$$

Nilai β dimaksimumkan didapatkan melalui turunan $L(\beta)$ terhadap β dan hasilnya adalah sama dengan nol

$$\frac{\partial L(\beta)}{\partial \beta_J} = \sum_{i=1}^{n} y_i x_{ij} - \sum_{i=1}^{n} x_{ij} \left(\frac{e^{\sum_{j=0}^{p}\beta_j x_{ij}}}{1+e^{\sum_{j=0}^{p}\beta_j x_{ij}}} \right)$$

Sehingga

$$\sum_{i=1}^{n} y_i x_{ij} - \sum_{i=1}^{n} x_{ij} \hat{\pi}(x_i) = 0 \qquad (2.8)$$

Dengan $j = 0,1,\ldots,p$

Estimasi varians dan kovarians dikembangkan melalui teori *MLE* dari koefisien parameternya (Rao, 1973 dalam Hosmer dan Lemeshow, 1989). Teori tersebut menyatakan bahwa estimasi varians kovarians didapatkan melalui turunan kedua $L(\beta)$.

$$\frac{\partial L(\beta)}{\partial \beta_j \beta_u} = \sum_{i=1}^{n} x_{ij} x_{iu} \pi(x_i)\left(1-\pi(x_i)\right); \text{dengan } j,u = 0,1,\ldots,p$$

Matriks varians kovarians berdasarkan estimasi parameter diperoleh malalui invers matriks dan diberikan sebagai berikut.

$$\hat{C}ov\left(\hat{\beta}\right)=\left\{x^T Diag\left[\hat{\pi}\left(x_i\right)\left(1-\hat{\pi}\left(x_i\right)\right)\right]x\right\}^{-1}$$

dimana $\boldsymbol{X}^T=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{11} & x_{21} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1k} & x_{2k} & \cdots & x_{nk} \end{bmatrix}$

Diagonal $\left[\hat{\pi}\left(x_i\right)\left(1-\hat{\pi}\left(x_i\right)\right)\right]$merupakan matriks diagonal (nxn) dengan diagonal utamanya adalah$\left[\hat{\pi}\left(x_i\right)\left(1-\hat{\pi}\left(x_i\right)\right)\right]$. Penaksir $SE(\beta)$ diberikan oleh akar kuadrat diagonal utama. Untuk mendapatkan nilai taksiran β dari turunan pertama fungsi $L(\beta)$ yang non linear maka digunakan metode iterasi *Newton Raphson*. Persamaan yang digunakan adalah.

$$\beta^{(t+1)}=\beta^{(t)}-\left(H^{(t)}\right)^{-1}q^{(t)};t=1,2,\ldots \tag{2.9}$$

dengan $q^T=\left(\frac{\partial L(\beta)}{\partial\beta_0},\frac{\partial L(\beta)}{\partial\beta_1},\ldots,\frac{\partial L(\beta)}{\partial\beta_k}\right)$ dan $\boldsymbol{H}$ merupakan matriks *Hessian*. Elemen-elemennya adalah $h_{ju}=\frac{\partial^2 L(\beta)}{\partial\beta_j\partial\beta_u}$,

sehingga $\boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & \cdots & x_{1k} \\ h_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ h_{k1} & h_{k2} & \cdots & x_{kk} \end{bmatrix}$. Pada setiap elemen berlaku hal berikut.

$$h_{ju}^{(t)} = \left.\frac{\partial^2 L(\beta)}{\partial\beta_j \partial\beta_u}\right|\beta^{(t)} = -\sum_{i=1}^{n} x_{ij} x_{iu} \pi(x_i)^{(t)} \left(1 - \pi(x_i)^{(t)}\right)$$

$$q_j^{(t)} = \left.\frac{\partial L(\beta)}{\partial\beta_j}\right|\beta^{(t)} = \sum_{i=1}^{n} \left(y_i - \pi(x_i)^{(t)}\right) x_{ij}$$

$$\pi(x_i)^{(t)} = \frac{e^{\left(\sum_{j=0}^{k} \beta_j^{(t)} x_{ij}\right)}}{1 + e^{\left(\sum_{j=0}^{k} \beta_j^{(t)} x_{ij}\right)}} \qquad (2.10)$$

dari persamaan di atas diperoleh persamaan berikut.

$$\beta^{(t+1)} = \beta^{(t)} + \left\{\boldsymbol{X}^{\boldsymbol{T}} Diag\left[\pi(x_i)^{(t)} \left(1 - \pi(x_i)^{(t)}\right)\right] x\right\}^{-1} \boldsymbol{X}^{\boldsymbol{T}} \left(y - \boldsymbol{m}^{(t)}\right) \qquad (2.11)$$

dengan $m^{(t)} = \pi(x_i)^{(t)}$. Langkah-langkah iterasi Newton Raphson diberikan sebagai berikut,

1. Menentukan nilai dugaan awal $\beta^{(0)}$ kemudian dengan menggunakan persamaan 2.10 maka didapatkan $\pi(x_i)^{(0)}$
2. Dari $\pi(x_i)^{(0)}$ pada langkah 1 diperoleh matriks *Hessian* $\boldsymbol{H}^0$ dan vektor $\boldsymbol{q}^0$

3. Proses selanjutnya untuk $t > 0$ digunakan persamaan 2.10 dan 2.11 hingga $\beta^{(t)}$ dan $\pi\left(x_i\right)^{(t)}$ konvergen.

2.2.2 Pengujian Estimasi Parameter

Untuk menguji signifikansi koefisien β dari model yang telah diperoleh, maka dilakukan uji parsial dan uji serentak. Setelah parameter hasil estimasi diperoleh, maka dilakukan pengujian keberartian terhadap koefisien β secara univariat terhadap variabel respon yaitu dengan membandingkan parameter hasil maksimum likelihood dugaan β dengan standar error parameter tersebut. Hipotesis pengujian parsial adalah sebagai berikut,

$H_0 : \beta_i = 0$

$H_1 : \beta_i \neq 0$ i=1,2, . . ., p

Statistik uji

$$W^2 = \frac{\beta_i^2}{SE(\beta_i)^2} \qquad (2.12)$$

Statistik uji tersebut mengikuti distribusi *chi-squared* sehingga H_0 ditolak ketika $W^2 > \chi^2_{(v,\alpha)}$dengan v adalah *degrees of freedom* banyaknya prediktor.

Setelah diperoleh variabel prediktor yang signifikan berpengaruh terhadap variabel respon pada pengujian univariat maka langkah selanjutnya adalah menentukan variabel manakah yang signifikan mempengaruhi variabel respon secara bersama-sama. Pengujian ini dilakukan untuk memeriksa keberartian koefisien β secara serentak (multivariat) terhadap variabel respon. Hipotesis yang digunakan adalah sebagai berikut,

$H_0 : \beta_1 = \beta_2 = \ldots = \beta_i = 0$

H_1: Paling tidak terdapat satu $\beta_i \neq 0; i = 1, 2, \ldots, p$

Satistik uji

$$G = -2\ln \frac{\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n_0}{n}\right)^{n_0}}{\sum_{i=1}^{n}\hat{\pi}^{y_i}\left(1-\hat{\pi}\right)^{(1-y_i)}} \qquad (2.13)$$

Dimana

$$n_i = \sum_{i=1}^{n} y_i \,;\; n_0 = \sum_{i=1}^{n}(1-y_i) \,;\; n = n_0 + n_1$$

Statistik uji G adalah merupakan *Likelihood Ratio Test* dimana nilai G mengikuti distribusi *Chi Squared* sehingga H_0 ditolak ketika $G > \chi^2_{(v,\alpha)}$dengan v adalah *degrees of freedom* banyaknya parameter dalam model tanpa β_0 (Hosmer dan Lomeshow,1989)

2.2.3 Interpretasi Koefisien Parameter

Interpretasi terhadap koefisien parameter ini dilakukan untuk menetukan kecenderungan/hubungan fungsional antara variabel prediktor dengan varibel respon serta menunjukkan pengaruh perubahan nilai pada variabel yang bersangkutan. Dalam hal ini digunakan nilai *odd ratio* atau e^{β} dan dinyatakan dengan ψ. *Odds ratio* diartikan sebagai kecenderungan variabel respon memiliki nilai tertentu jika diberikan x=1 dan dibandingkan pada x=0. Keputusan tidak ada hubungan antara variabel respond dan variabel prediktor diambil juka nilai *odd ratio* (ψ)=1

Jika nilai *odd ratio* (ψ)<1, maka antara variabel prediktor dan variabel respon terdapat hubungan negatif setiap kali perubahan nilai variabel bebas (x) dan jika *odds ratio* (ψ)>1 maka antara variabel prediktor dengan variabel respon terdapat hubungan positif setiap kali perubahan nilai variabel bebas (x) (Agresti, 1990).

2.2.4 Uji kesesuaian Model

Pengujian ini dilakukan untuk menguji apakah model yang dihasilkan berdasarkan regresi logistik multivariat atau serentak sudah layak. Dengan kata lain tidak terdapat perbedaan antara hasil pengamatan dengan kemungkinan hasil prediksi model.

Pengujian kesesuaian model dilakukan dengan hipotesis sebagai berikut.

H_0: Model sesuai (tidak terdapat perbedaan yang signifikan antara hasil pengamatan dengan kemungkinan hasil prediksi model)

H_1: Model tidak sesuai (terdapat perbedaan yang signifikan antara hasil pengamatan dengan kemungkinan hasil prediksi model)

Statistik uji

Perhitungan statistik uji *Chi-Square* sebagai berikut.

$$\chi^2 = \sum_{k=1}^{g} \frac{\left(o_k - n'_k \bar{\pi}_k\right)^2}{n'_k \bar{\pi}_k (1-\bar{\pi}_k)} \tag{2.14}$$

dimana

o_k $= \sum_{j=1}^{n'_k} y_j$ (jumlah pengamatan pada grup ke-*k*)

$\bar{\pi}_k$ $= \sum_{j=1}^{n'_k} \frac{m_j \pi_j}{n'_k}$ (rata-rata taksiran probabilitas)

g = jumlah grup (kombinasi kategori dalam model serentak)

m_j = banyaknya observasi yang memiliki nilai $\hat{\pi}_j$

n'_k = banyak observasi pada grup ke-*k*

Pengambilan keputusan tolak H_0 diambil ketika $\chi^2_{hitung} \geq \chi^2_{(db,\alpha)}$ dengan *db=g-2*. (Agresti, 1990)

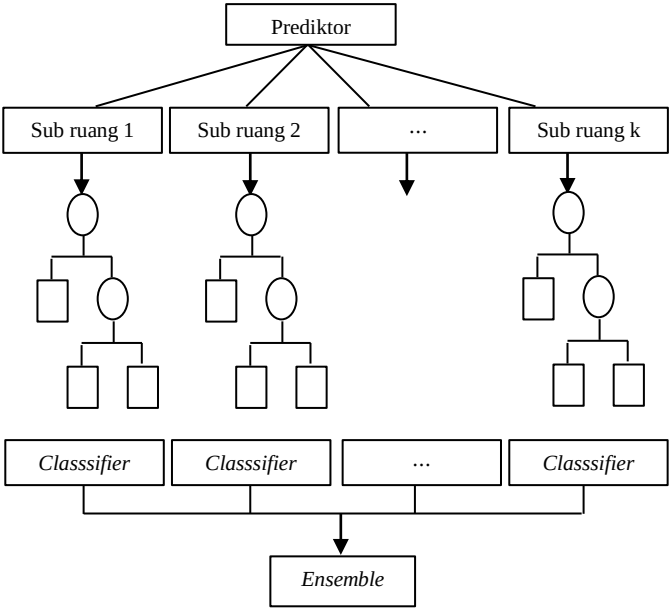
2.3 *Logistic Regression Classification by Ensembles from Random Partitions* (*LR*- CERP)

Pada LR-CERP, langkah pertama memilih Θ sebagai jarak antar variabel prediktor. Untuk meminimalkan korelasi di antara gabungan dari pengklasifikasi, Θ adalah random partisi kedalam *k* sub ruang $(\theta_1, \theta_2, \ldots, \theta_k)$ dengan ukuran yang sama. Sub ruang dipilih secara random dari distribusi yang sama maka pemilihan variabel diasumsikan tidak mengalami bias antar masing-masing sub ruang. Pada masing-masing sub ruang digunakan model regresi logistik tanpa pemilihan variabel. Diharapkan akan

didapatkan probabilitas error yang hampir sama dari *k* pengklasifikasi dan didapatkan peningkatan akurasi.

CERP mengkombinasikan hasil model *multiple* regresi logistik untuk meningkatkan akurasi dari prediksi menggunakan mayoritas *voting* dari pengklasifikasi atau dari gabungan rata-rata nilai prediksi. Untuk meningkatkan performa CERP lebih lanjut maka diselidiki mayoritas *voting* diantara kumpulan dari *ensemble*. Skema untuk random partisi dengan algoritma CERP digambarkan pada Gambar 2.1

Untuk meningkatkan keseimbangan antara akurasi maka LR-CERP menggunakan optimal *threshold* untuk diklasifikasikan berdasarkan pengklasifikasi. Jika penetapan nilai *threshold* sebesar 0,5 memberikan hasil klasifikasi yang buruk karena jumlah respon data tidak seimbang maka digunakan nilai *threshold r* sebagai proporsi respon positif pada data set. Jika nilai *fitted value* melebihi nilai *r* maka sampel diklasifikasikan sebagai 1 dan diklasifikasikan kedalam 0 jika nilai *fitted value* kurang dari nilai *r*.



Gambar 2.1 Ilustrasi Pohon Klasifikasi

Tingkat respon positif (*r*) belum tentu merupakan optimal *threshold* dalam hal menyeimbangkan *sensitivity* dan *spesifisity Threshold* optimal biasanya terletak di antara 0,5 dan tingkat respon positif (*r*). Peningkatan yang substansial dalam menyeimbangkan *sensitivity* dan *spesifisity* untuk LR-CERP menggunakan *threshold* yang berbeda dari 0,5 untuk data yang tidak seimbang (Ahn, Moon, Fazzari, Lim, Chen, & Kodell, 2006). Performa CERP tergantung pada banyaknya prediktor pada satu partisi yang ditentukan oleh banyaknya partisi optimal. Partisi optimal didapatkan dari persamaan berikut.

$$K = \frac{6 \times p}{n} \qquad (2.15)$$

dimana p adalah banyaknya prediktor dan n adalah banyaknya pengamatan. Apabila ukuran n lebih besar dari ukuran p, maka partisi optimal dapat diperoleh dengan membagi data sebanyak i

menjadi $\frac{n}{i}$ dimana i adalah sembarang integer yang kurang dari n. Relatifnya partisi optimal didapatkan dari $K = \frac{n}{i}$ akurasi yang menghasilkan akurasi tertinggi. (Lim, 2007).

2.4 *Logistic Regression Ensembles* (*Lorens*)

Berdasarkan algoritma CERP (*Classification by Ensemble from Random Partition*) dikembangkan metode *Lorens* dengan menggunakan model regresi logistik sebagai pengklasifikasi. *Lorens* mengkombinasikan nilai prediksi dari model *multiple* regresi logistik untuk meningkatkan akurasi klasifikasi dengan hasil rata-rata pada suatu *ensemble.* Nilai prediksi pada gabungan pengklasifikasi (model regresi logistik) adalah nilai rata-rata dan diklasifikasikan kedalam 0 dan 1 menggunakan nilai *threshold* optimal.

Lorens mengulangi prosedur yang digunakan LR CERP beberapa kali untuk digabungkan dalam satu *ensemble* hingga terbentuk beberapa *ensemble.* Seperti halnya pada LR CERP, *Lorens* mampartisi data kedalam *k* sub ruang yang dipilih secara random dari distribusi yang sama sehingga pemilihan variabel diasumsikan tidak mengalami bias antar masing-masing sub ruang. Pada masing-masing sub ruang digunakan model regresi logistik tanpa pemilihan variabel. Diharapkan akan didapatkan probabilitas error yang hampir sama dari *k* pengklasifikasi dan didapatkan peningkatan akurasi.

Thershold yang biasa digunakan dalam klasifikasi untuk respon biner adalah 0.5. Namun akurasi klasfikasi tidak akan baik apabila proporsi kelas 1 dan 0 tidak seimbang. Dalam rangka menyeimbangkan *sensitivity* dan *specificity*, *Lorens* mencari *threshold* optimal melalui rumus berikut.

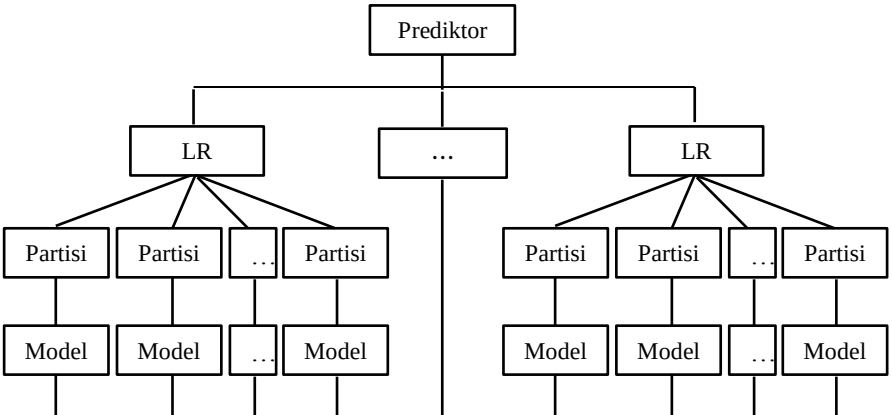
$$Threshold = \frac{r + 0.5}{2} \tag{2.16}$$

Dimana *r* merupakan proporsi respon positif yang terdapat pada data.

Meskipun sebagian besar metode mayoritas *voting* dan rata-rata adalah sama namun penggunaan metode *Lorens* pada

beberapa kasus memeberikan hasil prediksi yang lebih tinggi. *Lorens* menghasilkan beberapa *ensemble* dengan partisi random yang berbeda-beda dan memilih nilai terbanyak di antara beberapa *ensemble.* Dari nilai tersebutlah didapatkan satu akurasi umum yang akurasinya telah ditingkatkan dari kontribusi beberapa *ensemble* yang ditentukan. Peningkatan akurasi ini akan didapatkan jika jumlah *ensemble* yang dibangun lebih dari 10 (Lim, Ahn, Moon, & Chen,2010). Karena partisi acak, *Lorens* bebas dari masalah dimensi data. Pemilihan variabel tidak diperlukan di *Lorens* dan korelasi antar klasifikasi dapat dikurangi melalui partisi acak. Daripada CERP, *Lorens* unggul dalam basis klasifikasi yang menggunakan regresi logistik yang popular dan mudah dipahami. Selain itu *Lorens* juga lebih efisien dalam hal komputasi dibandingkan CERP yang menggunakan algoritma pohon (*tree*). Dengan mengintegrasikan keuntungan ini menuju pemilihan *ensemble* umum, keakuratan metode dapat menjadi lebih baik. Prosedur *Lorens* melibatkan LR CERP yang dilakukan berulang-ulang. Berikut ini merupakan diagram alir prosedur *Lorens* (Lee dkk, 2013). Ilustrasi pengulangan prosedur LR CERP pada metode *Lorens* terdapat pada Gambar 2.2.

Gambar 2.2. menunjukkan alur yang digunakan untuk menghitung ketepatan klasifikasi akhir pada metode *Lorens*. Langkah awal adalah mempartisi data secara random kemudian akan dibentuk model dari masing-masing partisi. Probabilitas yang didapatkan dari masing-masing partisi dalam 1 *ensemble* akan dirata-rata, dan diperoleh probabilitas akhir pada masing-masing *ensemble*. Selanjutkan akan dilakukan mayoritas *voting* dari masing-masing *ensemble* dan didapatkan ketepatan klasifikasi akhir.



Gambar 2.2 Ilustrasi Metode *Lorens*

Menurut Johnson (2007) untuk menghitung ketepatan klasifikasi pada hasil pengelompokkan digunakan *apparent error rate* (APER). Nilai APER menyatakan representasi proporsi sampel yang salah diklasifikasikan. Dalam penelitian ini digunakan respon biner dua kategori sehingga penentuan kesalahan klasifikasi dapat dihitung dari tabel klasifikasi berikut.

Tabel 2.1 Klasifikasi Respon Biner Dua Kategori

Aktual	Prediksi		Total
	1	2	
1	n11	n12	N1

2	n21	n22	N2
Total	N1	N2	N

Ketepatan klasifikasi dapat dijelaskan melalui Nilai *sensitivity* dan *specificity* dan *APER*. *Sensitivity* merupakan kemampuan untuk menebak respon positif dan *specificity* merupakan kemampuan untuk menebak respon negatif. Berikut ini adalah rumus untuk *sensitivity*, *specificity* dan APER

$$sensitivity = \frac{n11}{n11 + n21} \tag{2.17}$$

$$specificity = \frac{n11}{n11 + n21} \tag{2.18}$$

$$APER(\%) = \frac{n_{12} + n_{21}}{N} x100\% \tag{2.19}$$

$$Ketepatan\ klasifikasi\ = 1 - APER \tag{2.20}$$

Keterangan:

n_{21} : jumlah observasi dari kelas 2 yang salah diprediksi sebagai kelas 1

n_{12} : jumlah observasi dari kelas 1 yang salah diprediksi sebagai kelas 2

n_{22} : jumlah observasi dari kelas 2 yang tepat diprediksi sebagai kelas 2

$N_{1.}$: jumlah observasi dari kelas 1

$N_{2.}$: jumlah observasi dari kelas 2

N : jumlah observasi (Martono, 2014)

2.5 *Cross Validation*

Prosedur yang dapat diterapkan untuk mengevaluasi performa model dan melalui data *training* dan data *testing* adalah metode *holdout* dan *cross validation*. Metode *holdout* menggunakan data dengan jumlah tertentu sebagai data *training* dan sebagian data sisanya sebagai data *testing*. Namun pada umumnya dua-pertiga dari data digunakan sebagai data *training* dan sepertiga dari data diguanakan sebagai data *testing*. Sampel yang diguanakan sebagai data *training* dan *testing* mungkin saja

tidak representatif. Namun secara umum tidak dapat dikatakan apakah data yang digunakan sebagai data *training* dan *testing* sudah representatif. Suatu cek sederhana yang dapat dilakukan agar data *training* dan data *testing* representatif yaitu dengan cara memastikan bahwa setiap kelas dalam dataset penuh harus terwakili dalam proporsi yang tepat untuk *training* dan *testing*. Jika semua sampel dengan kelas tertentu dihilangkan dari *training* set, *classifier* tidak dapat diharapkan belajar dengan baik dari data yang tersedia dalam melakukan klasifikasi pada *testing set*. Maka harus dipastikan bahwa pengambilan sampel dilakukan dengan cara random yang menjamin bahwa setiap kelas terwakili baik pada *training* dan *testing set*. Prosedur seperti ini dinamakan dengan stratifikasi, namun stratifikasi hanya menyediakan perlindungan yang lemah terhadap kelas yang tidak representatif dalam *training* dan *testing set*. Secara umum untuk mengurangi bias yang diakibatkan oleh pengambilan sampel pada metode *holdout* yaitu dengan cara mengulang seluruh proses pada data *training* dan *testing* beberapa kali dengan sampel secara acak yang berbeda-beda. Dalam setiap iterasi dengan proporsi tertentu, dua-pertiga dari data yang dipilih secara acak untuk data *training* dengan stratifikasi dan sisanya digunakan untuk data *testing*. Tingkat kesalahan pada iterasi yang berbeda dirata-ratakan untuk menghasilkan tingkat kesalahan keseluruhan. Metode ini secara sederhana dikenal sebagai metode *holdout* berulang untuk mengestimasi tingkat kesalahan.

Pada prosedur *holdout* tunggal dapat dilakukan penukaran antara data *training* dan data *testing* untuk melatih sistem dan untuk melatih sistem dan hasil dari dua proses tersebut dirata-rata untuk mengurangi efek dari data yang tidak representatif pada data *training* dan *testing* namun hal ini hanya dapat dilakukan ketika proporsi pembagian data sebesar 50:50 namun pada umumnya lebih baik digunakan lebih dari setengah dari data set untuk sebagai data *training*. Sehingga digunakan *cross validation*, pada *cross validation* data dibagi menjadi beberapa *k folds* atau partisi yang sama banyak, setiap *folds* pada gilirannya digunakan

untuk *testing* dan sisanya digunakan untuk *training*. Misalkan digunakan partisi sebanyak tiga, maka data akan dipartisi menjadi tiga bagian yang sama, masing-masing pada gilirannya digunakan untuk data *training* dan sisanya sebagai data *testing*. Artinya, menggunakan dua-pertiga dari data untuk data *training* dan sepertiga untuk data *testing*, dan ulangi prosedur tiga kali maka hal ini disebut *threefold cross-validation* dan jika stratifikasi diadopsi maka disebut *stratified threefold cross-validation.*

Cara standar memprediksi tingkat kesalahan adalah menggunakan *stratified 10-fold cross-validation.* Pada *10-fold cross-validation* data dibagi menjadi 10 bagian dengan proporsi yang sama, maka sembilan-persepuluh dari data untuk data *training* dan sepersepuluh untuk data *testing*, dan ulangi prosedur sebanyak 10 kali. Akhirnya, 10 prediksi kesalahan dirata-rata untuk menghasilkan prediksi kesalahan secara keseluruhan. Tes pada berbagai dataset yang berbeda menunjukkan bahwa partisi 10 bagian data adalah yang tepat untuk mendapatkan estimasi kesalahan yang terbaik. Terdapat beberapa bukti teori yang mendukung hal ini. Meskipun argumen ini tidak berarti konklusif di kalangan *Machine Learning* dan *Data Mining* tentang apa skema terbaik untuk evaluasi, 10 *folds cross validation* telah menjadi metode standar dalam praktis. Tes juga menunjukkan bahwa penggunaan stratifikasi meningkatkan ketepatan prediksi.

10 *folds cross validation* mungkin tidak cukup mampu mendapatkan estimasi kesalahan yang handal karena sering menghasilkan hasil yang berbeda karena efek variasi acak dalam memilih *fold*. Stratifikasi dapat mengurangi variasi, tapi tentu tidak menghilangkannya sama sekali. Ketika mencari prediksi kesalahan yang akurat, adalah prosedur standar untuk mengulangi proses *cross-validation* 10 kali dan melakukan rata-rata terhadap hasilnya. Prosedur ini melibatkan 100 kali algoritma pada dataset, dibutuhkan usaha komputasional yang intensif untuk mendapatkan performa ukuran yang baik. Pembagian data menjadi sembilan-persepuluh bagian sebagai data *training* dan

sepersepuluh bagian sebagai data *testing*, dapat diadopsi pada prosedur *holdout* yang digunakan untuk menguji kehandalan model dalam memprediksi kelas pengamatan. (Witten, Frank, dan Hall, 2011).

2.6 Enzim

Enzim adalah protein yang dapat meningkatkan laju reaksi kimia baik dengan pembuatan maupun pemecahan ikatan kovalen, dimana ligannya dinamakan substrat. Walaupun tidak semua protein adalah enzim, namun enzim merupakan kelas yang cukup besar dan penting dalam protein karena hampir semua reaksi kimia pada sel dikatalisasi oleh enzim yang spesifik (Lodish, et al, 2008). Senyawa apapun yang dapat mengurangi kecepatan dari reaksi yang dikatalis enzim dinamakan inhibitor. Inhibitor ireversibel berikatan dengan enzim melalui ikatan kovalen (Champe and Harvey, 2007)

2.7 *Docking*

Di bidang *molecular modeling*, *docking* merupakan metode yang memprediksi orientasi yang disukai dari suatu molekul ketika *binding* dengan molekul lain untuk membentuk kompleks stabil (Lengauer dan Rarey, 1996). *Docking* paling sering digunakan untuk memprediksi orientasi *binding* kandidat obat molekul kecil dengan target proteinnya untuk memprediksi afinitas dan aktivitas molekul kecil.

2.8 *In Silico*

Sintetis tradisional sejumlah senyawa baru menggunakan cara konvensional memakan waktu dan biaya yang besar, di sisi lain penapisan secara *in silico* memberi alternatif baru. Penapisan *in silico* lebih efektif dibandingkan dengan penapisan *in vitro* dan *in vivo*. Penapisan *in silico* dalam kaitannya dengan penyakit dilakukan untuk menemukan inhibitor potensial. Kelebihan dari penapisan *in silico* adalah kemampuannya untuk membedakan senyawa aktif dan inaktif sehingga hal ini dapat menghemat waktu dan sumber daya lainnya (Kirchmair, Markt, Distinto, Wolber & Langer, 2008).

2.9 *Database DUD-E*

DUD-E merupakan *benchmarking database* untuk simulasi *docking*. *Database* ini berisi tiga tipe data yaitu target enzim, *ligand* dan *decoy*. *DUD-E* teridiri dari 102 jenis enzim dengan banyak *ligand* dan *multiple* strukturnya serta terdapat *ligand* dari *chemical database ChEMBL*, *ligand* tersebut dikelompokkan oleh *ChEMBL ID*. Pada masing-masing kelompok, *ligand* mempunyai struktur yang sama namun mempunyai sumber yang berbeda-beda. Kebanyakan dari *ligand* mempunyai struktur yang hampir sama sehingga dibutuhkan ketelitian untuk mengelompokkan *ligand*. *Database DUD-E* mempunyai koleksi *decoys* dari senyawa pada *Database ZINC* berdasarkan pada ligand tersebut. *Decoy* mempunyai properti yang sama dengan *ligand* namun keduanya mempunyai struktur yang berbeda, oleh karena itu diasumsikan bahwa *decoy* adalah inhibitor buruk. Berikut ini adalah tabel senyawa penyusun dari enzim pada *Database DUD-E* dengan masing-masing jumlah *ligand* dan *decoy* pada masing-masing enzim.

Tabel 2.2 Senyawa Penyusun Enzim

Enzim	Deskripsi	*Ligand*	*Decoy*
Aofb	*Manoamine oxidase B*	168	504
Cah2	*Carbonic Anhidrase II*	835	2505
Hs90a	*Heat stock protein HSP 90-alpha*	125	375

Berdasarkan Tabel 2.2 diketahui bahwa enzim *aofb* terdiri dari 168 *ligand* dan 504 *decoy*, enzim *cah2* terdiri dari 835 *ligand* dan 2505 *decoy* serta enzim *hs90a* terdiri dari 125 *ligand* dan 375 *decoy*.

2.9.1 *Aofb* (*Manoamine oxidase B)*

Monoamine oxidase merupakan suatu sistem enzim kompleks yang terdistribusi luas dalam tubuh, berperan dalam dekomposisi amin biogenik, seperti norepinefrin, epinefrin, dopamine, serotonin. MAOI menghambat sistem enzim ini,

sehingga menyebabkan peningkatan konsentrasi amin endogen. Terdapat dua tipe MAO yang telah teridentifikasi, yaitu MAO-A dan MAO-B. Kedua enzim ini memiliki substrat yang berbeda serta perbedaan dalam sensitivitas terhadap inhibitor. MAO-A cenderungan memiliki aktivitas deaminasi epinefrin, norepinefrin, dan serotonin, sedangkan MAO-B memetabolisme benzilamin dan fenetilamin. Dopamin dan tiramin dimetabolisme oleh kedua isoenzim.

Pada jaringan syaraf, sistem enzim ini mengatur dekomposisi metabolik katekolamin dan serotonin. MAOI hepatic menginaktivasi monoamin yang bersirkulasi atau yang masuk melalui saluran cerna ke dalam sirkulasi portal (misalnya tiramin). Semua MAOI nonselektif yang digunakan sebagai antidepresan merupakan inhibitor ireversibel, sehingga dibutuhkan sampai 2 minggu untuk mengembalikan metabolisme amin normal setelah penghentian obat. Hasil studi juga mengindikasikan bahwa terapi MAOI kronik menyebabkan penurunan jumlah reseptor (*down regulation*) adrenergik dan serotoninergik (Departemen Kesehatan RI, 2007).

2.9.2 *Cah2 (Carbonic Anhidrase II)*

Carbonic Anhidrase merupakan enzim logam seng yang mengkatalisis reaksi reversibel: $CO_2 + H_2O \leftrightarrow H_2{+}CO_2 \leftrightarrow HC0_3{}^- {+} H^+$. Reaksi ini membentuk dasar untuk regulasi keseimbangan asam-basa dalam organisme. CA2 adalah satu-satunya bentuk terlarut karbonat anhidrase di sel epitel ginjal. Di sisi lain, CA4 dinyatakan baik pada apikal perbatasan membran dan membran basolateral dari sel tubulus proksimal. Ginjal menyerap semua bikarbonat yang disaring oleh glomeruli. Kebanyakan, 70%-85% dari bikarbonat diserap kembali di tubulus proksimal dan hanya 10%-20% di diserap pada Henle. Pada segmen tersebut, intraseluler CA2 dan apikal CA4 keduanya bertanggung jawab atas transepitelial bersih transportasi bikarbonat (Lin, Liao, Horng, Yan, Hsiao, Hwang, 2008)

2.9.3 *Hs90a (Heat shock protein HSP 90-alpha)*

Heat shock protein telah diketahui memiliki peran penting terhadap resistensi insulin. Resistensi insulin merupakan salah satu penyebab terjadinya diabetes melitus, yakni penyakit yang memiliki angka mortalitas tertinggi keempat secara global menurut WHO. Resistensi insalin dipengaruhi oleh berbagai faktor salah satunya adalah keberadaan asam lemak bebas. Asam lemak bebas mempengaruhi fosforilasi reseptor insalin melalui mekanisme yang melibatkan I-K-Kinase-β dan Jun Nkinase (JNK). *Heat shock protein* berperan dalam mencegah fosforilasi reseptor insulin. Fosforilasi ini mengakibatkan inaktivasi reseptor. Selain itu Hsp juga mempengaruhi mekanisme pertahanan sel islet terhadap radikal bebas dan menurunkan oksidasi lipid, yang juga berkaitan dengan patogenesis resistensi insulin. Resistensi insulin dapat diperbaiki dengan menggunakan terapi-terapi yang berfungsi meningkatkan ekspresi Hsp. (Widjaja, Santoso, & Waspadji, 2009)

2.9.4 Senyawa Penyusun Enzim

Senyawa yang berasal dari *Database DUD-E* diinputkan ke *Discovery Studio*. *Discovery Studio* merupakan *3D* modeling *studio* yang meliputi banyak fungsi untuk *discovery* obat. Setelah senyawa diinputkan maka dapat dihitung molecular properti berdasarkan protocol yang terdapat pada *Discovery Studio*. Protocol tersebut meliputi 3 properti yaitu tipe *1D,2D* dan *3D.* Protocol *1D* meliputi *specific atoms,* protocol *2D* meliputi *AlogP* dan protocol *3D* meliputi *Surface Area*, yang dihitung menggunakan struktur *3D*. Berikut ini adalah beberapa properti yang digunakan pada *Discovery Studio* yang merupakan tipe dari penyusun senyawa enzim.

Tabel 2.3. Properti pada *Discovery Studio*

Tipe	Properti	Type	Properti
A	*ALogP*	C	*NPlusO_Count*
A	*ALogP_MR*	C	*Num_Atoms*
A	*ALogP98*	C	*Organic_Count*
W	*Molecular_Weight*	E	*Energy*

W	*Molecular_Mass*	E	*Minimized_Energy*
W	*Molecular_Solubility*	E	*Strain_Energy*
W	*VSA_TotalArea*	O	*FormalCharge*
C	*HBA_Count*	O	*IsChiral*
C	*HBD_Count*	O	*AverageBondLength*

Keterangan
A : Tipe *A Log P*
C : Tipe struktur Spesifik
W : Tipe berat permukaan
E : Tipe energi
O : Tipe lainnya

(*Halaman ini sengaja dikosongkan***)**

BAB III
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan pada penelitian ini adalah data sekunder mengenai yaitu struktur enzim dengan simulasi *docking* yang diklasifikasikan kedalam jenis *ligand* dan *decoy*. Data sekunder tentang enzim di dapatkan dari *database DUD-E* yang meliputi 102 jenis enzim yang terdiri dari masing-masing struktur enzim namun pada penelitian ini hanya digunakan 3 jenis enzim yang terdiri dari target enzim, *ligand* dan *decoy*.

3.2 Variabel Penelitian

Pada penelitian ini digunakan 3 jenis enzim yaitu *aofb*, *cah2 dan hs90a* sehingga terdapat 3 variabel penelitian. Variabel prediktor terdiri dari senyawa penyusun enzim dan variabel respon terdiri dari klasifikasi enzim yang terdiri dari inhibitor baik (*ligan*) dan inhibitor buruk (*decoy*). Berikut ini adalah variabel penelitian untuk 3 jenis enzim.

Tabel 3.1 Variabel Penelitian

Enzim	Variabel	Skala
	Klasifikasi enzim y(0)=*Decoy* (Inhibitor buruk) y(1)=*Ligand* (Inhibitor baik)	Nominal
aofb	x_1 =*A Log P* x_2 =*A Log P MR* x_3 =*A Log P 98* ⋮ x_{69} =*Molecular_3D_PolarSASA* x_{70} =*Molecular_3D_SAVol*	Rasio
cah2	Klasifikasi enzim y(0)=*Decoy* (Inhibitor buruk) y(1)=*Ligand* (Inhibitor baik)	Nominal

Tabel 3.1 Variabel Penelitian (*Lanjutan*)

Enzim	Variabel	Skala
	x_1 =*A Log P* x_2 =*A Log P MR* x_3 =*A Log P 98* ⋮ x_{69} =*Molecular_3D_PolarSASA* x_{70} =*Molecular_3D_SAVol*	Rasio
	Klasifikasi enzim y(0)=*Decoy* (Inhibitor buruk) y(1)=*Ligand* (Inhibitor baik)	Nominal
hs90a	x_1 =*A Log P* x_2 =*A Log P MR* x_3 =*A Log P 98* ⋮ x_{69} =*Molecular_3D_PolarSASA* x_{70} =*Molecular_3D_SAVol*	Rasio

3.3 Struktur Data Penelitian

Data yang digunakan pada penelian ini adalah data enzim yang mengandung beberapa jenis senyawa penyusun. Data terdiri dari 3 jenis enzim dimana masing-masing enzim terdiri dari senyawa penyusun dan jumlah pengamatan yang berbeda. Berikut ini adalah struktur data untuk masing-masing jenis enzim.

Tabel 3.2 Struktur Data Penelitian

Jenis Enzim	Pengamatan	Variabel prediktor				Variabel respon
		$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	⋯	$\boldsymbol{x_{70}}$	$\boldsymbol{y}$
aofb	Pengamatan 1	x_{11}	x_{12}	⋯	x_{170}	y_1
	Pengamatan 2	x_{21}	x_{22}	⋯	x_{270}	y_2
	⋮	⋮	⋮	⋮	⋮	⋮
	Pengamatan 672	x_{n1}	x_{n2}	⋯	x_{n70}	y_{672}

Tabel 3.2 Struktur Data Penelitian(*Lanjutan*)

Jenis Enzim	Pengamatan	Variabel prediktor				Variabel respon
cah2		$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\cdots$	$\boldsymbol{x_{71}}$	$\boldsymbol{y}$
	Pengamatan 1	x_{11}	x_{12}	$\cdots$	x_{171}	y_1
	Pengamatan 2	x_{21}	x_{22}	$\cdots$	x_{271}	y_2
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	Pengamatan 3340	x_{n1}	x_{n2}	$\cdots$	x_{n71}	y_{3340}
hs90a		$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\cdots$	$\boldsymbol{x_{69}}$	$\boldsymbol{y}$
	Pengamatan 1	x_{11}	x_{12}	$\cdots$	x_{169}	y_1
	Pengamatan 2	x_{21}	x_{22}	$\cdots$	x_{269}	y_2
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	Pengamatan 500	x_{n1}	x_{n2}	$\cdots$	x_{n69}	y_{500}

Pada struktur data subjek yang digunakan adalah pengamatan pada penyusun senyawa enzim. Enzim *aofb* terdiri dari 672 pengamatan dengan 70 variabel prediktor, enzim *cah2* terdiri dari 3340 pengamatan dengan 71 variabel prediktor dan nzim *hs90a* terdiri dari 500 pengamatan dengan 69 variabel prediktor.

3.4 Metode Analisis

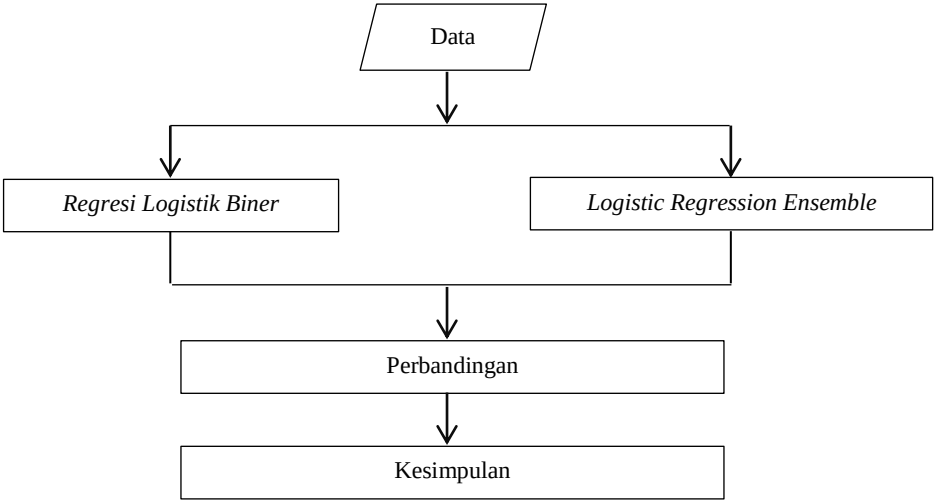
Analisis yang digunakan dalam penelitian ini adalah analisis regresi logistik biner dan *logistic regression ensemble* (*Lorens*) yang digunakan untuk pengklasifikasian senyawa enzim pada database *DUD-E*. Berikut tahapan analisis dalam penelitian ini.

1. Membagi data kedalam 90% dan 85% data *training* serta 10% dan 15% data *testing*
2. Melakukan analisis regresi logistik biner melalui tahapan berikut.
   a) Mengestimasi parameter dengan *maximum likelihood*
   b) Menguji parameter secara parsial dan simultan
   c) Menginterpetasi *Odd Ratio*
   d) Menyusun model regresi logistik

e) Melakukan transformasi logit model regresi logistik
f) Memprediksi klasifikasi dan menghitung nilai akurasi
g) Menghitung ketepatan klasifikasi 10% data *testing* dari persamaan regresi logistik yang terbentuk
h) Membandingkan hasil ketepatan klasifikasi dari setiap kombinasi data *training* dan *testing.*

3. Melakukan analisis *logistic regression ensemble* (*Lorens*) dengan tahapan sebagai berikut
   a) Menentukan jumlah partisi (*k*) yang akan dicobakan dalam penelitian dan jumlah *ensemble* (*n*)
   b) Mempartisi variabel kedalam *k subspace*
   c) Membentuk model regresi logistik pada masing-masing partisi
   d) Menghitung nilai probabilitas dari model yang terbentuk pada setiap partisi
   e) Menghitung rata-rata nilai probabilitas dari setiap partisi dalam satu *ensemble*
   f) Mengklasifikasikan setiap probabilitas pada satu ensemble kedalam respon positif atau respon negatif
   g) Melakukan mayoritas *voting* dari 10 *ensemble*
   h) Mendapatkan ketepatan klasifikasi
   i) Mengulang langkah pada a sampai h dengan jumlah partisi (2,3,4,5,6,7,8,9 dan 10) dan dipilih satu partisi yang optimal
   j) Membandingkan hasil ketepatan klasifikasi dari setiap kombinasi data *training* dan *testing.*
4. Melakukan analisis *logistic regression ensemble* (*Lorens*) menggunakan evaluasi 10 *fold cross validation.*
5. Membandingkan tingkat akurasi klasifikasi antara regresi logistik biner dengan *logistic regression ensemble* (*Lorens*) dengan partisi yang optimal. Klasifikasi yang mempunyai 1-APER terbesar dipilih sebagai tingkat klasifikasi yang lebih baik dalam melakukan klasifikasi.
6. Menarik kesimpulan.

Diagram analisis langkah penelitian digambarkan seperti pada Gambar 3.1 berikut.



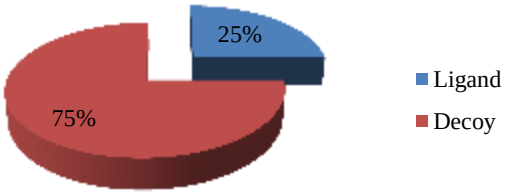
Gambar 3.1 Diagram Alir

(*Halaman ini sengaja dikosongkan***)**

BAB IV
ANALISIS DAN PEMBAHASAN

4.1 Karakteristik Enzim Pada *Database DUD-E*

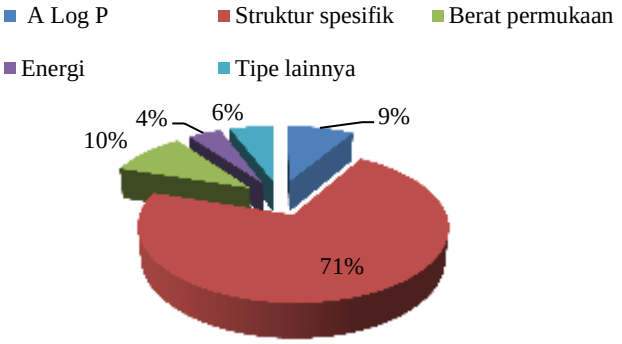
Pada pembahasan ini dianalisis tiga jenis enzim pada database *DUD-E* yaitu enzim *aofb* yang merupakan *Monoamine oxidase*, enzim *cah2* yang merupakan *Carbonic Anhidrase* dan enzim *hs90a* yang merupakan *Heat shock protein*. Masing-masing mempunyai karakteristik yang berbeda-beda. Enzim *aofb* terdiri dari 672 pengamatan dengan 70 senyawa penyusun, enzim *cah2* terdiri dari 3340 pengamatan dengan 71 senyawa penyusun dan enzim *hs90a* terdiri dari 500 pengamatan dengan 69 senyawa penyusun. Setiap pengamatan pada masing-masing-masing enzim terdiri dari 2 respon yaitu inhibitor baik (*ligand*) dan ihibitor buruk (*decoy*). Berikut ini adalah proporsi untuk kedua jenis variabel respon pada masing-masing enzim *aofb, cah2* dan *hs90a*



Gambar 4.1 Proporsi *ligand* dan *decoy* pada enzim *aofb,cah2* dan *hs90a*

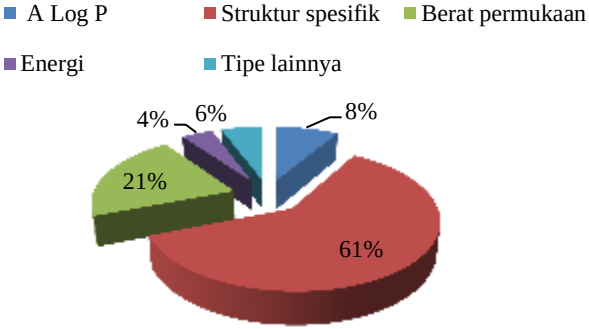
Gambar 4.1 menunujukkan presentase respon positif pada enzim *aofb* adalah sebesar 25 persen atau sebanyak 168 pengamatan sedangkan presentase respon negatif atau *decoy* adalah sebesar 75 persen atau sebanyak 504 pengamatan. Proporsi respon yang sama juga terdapat pada enzim *cah2* dan *hs90a*. Pada enzim *cah2 proporsi* respon positif atau *ligand* adalah sebesar 25 persen atau sebanyak 835 pengamatan sedangkan proporsi respon negatif atau *decoy* adalah sebesar 75 persen atau sebanyak 2505 pengamatan. Pada enzim *hs90a proporsi* respon positif atau *ligand* adalah sebesar 25 persen atau sebanyak 125 pengamatan

sedangkan proporsi respon negatif atau *decoy* adalah sebesar 75 persen atau sebanyak 375 pengamatan. Ketiga enzim mempunyai lima tipe senyawa penyusun yaitu *A Log P*, struktur spesifik, berat permukaan, energi dan tipe lainnya. Berikut ini adalah proporsi dari kelima tipe senyawa tersebut pada enzim *aofb*



Gambar 4.2 Proporsi Tipe Senyawa Pada Enzim *aofb*

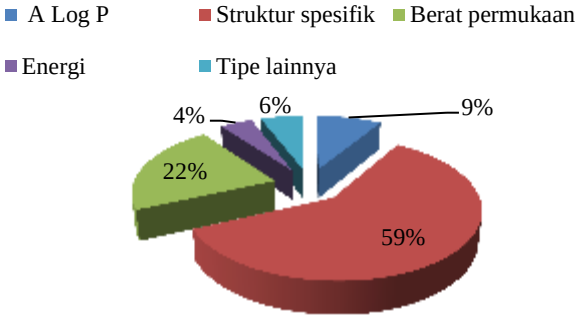
Berdasarkan grafik pada Gambar 4.2 dapat diketahui bahwa 71 persen senyawa atau variabel prediktor pada enzim *aofb* terdiri dari tipe struktur spesifik, 21 persen terdiri dari berat permukaan, 9 persen terdiri dari *A Log P*, 4 persen dari unsur energi dan 6 persen dari tipe lainnya. Berikut ini adalah proporsi dari kelima tipe senyawa pada enzim *cah2*.



Gambar 4.3 Proporsi Tipe Senyawa Pada Enzim *cah2*

Gambar 4.3 menunjukkan proporsi tipe senyawa pada enzim *cah2*. Berdasarkan gambar tersebut dapat diketahui bahwa

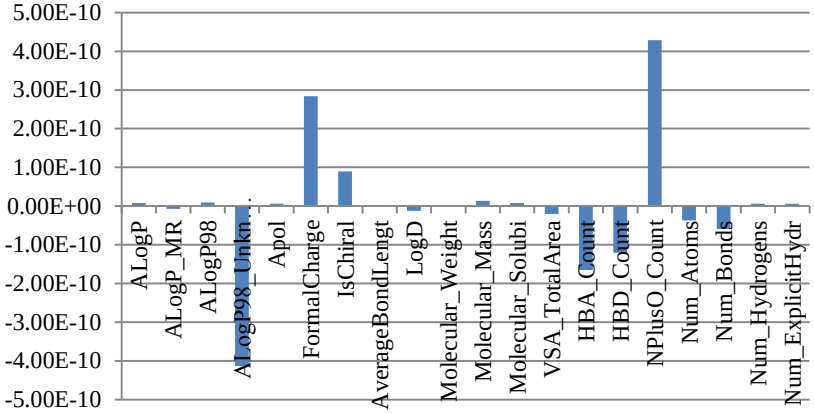
dapat diketahui bahwa 61 persen senyawa atau variabel prediktor pada enzim *cah2* terdiri dari tipe struktur spesifik, 21 persen terdiri dari berat permukaan, 8 persen terdiri dari *A Log P*, 4 persen dari unsur energi dan 6 persen dari tipe lainnya. Berikut ini adalah proporsi dari kelima tipe senyawa pada enzim *hs90a.*



Gambar 4.4 Proporsi Tipe Senyawa Pada Enzim *hs90a*

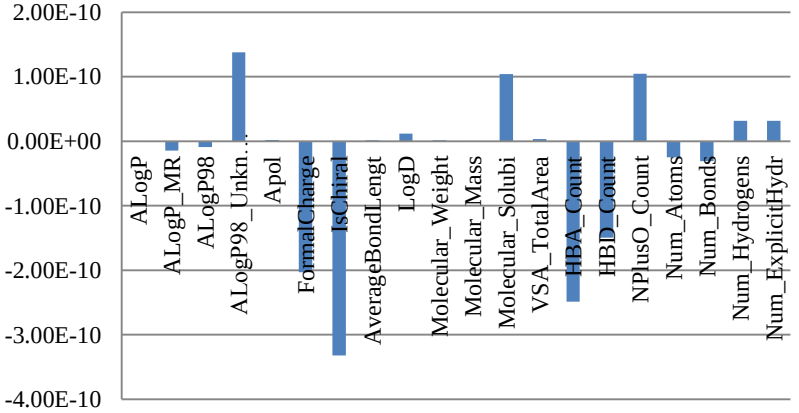
Gambar 4.4 menunjukkan bahwa 59 persen senyawa atau variabel prediktor pada enzim *hs90a* terdiri dari tipe struktur spesifik, 22 persen terdiri dari berat permukaan, 9 persen terdiri dari *A Log P*, 4 persen dari unsur energi dan 6 persen dari tipe lainnya. Karakteristik enzim juga dapat dilihat dari sebaran rata-rata pada masing-masing enzim. Grafik yang menunjukkan sebaran rata-rata 20 senyawa pada enzim *aofb* terdapat pada Gambar 4.5.

Gambar 4.5 menunjukkan sebaran nilai rata-rata kandungan 20 senyawa penyusun enzim *aofb.* Terdapat rata-rata senyawa yang berada diatas nol dan berada dibawah nol. Senyawa-senyawa dengan kandungan senyawa bernilai dibawah nol cenderung memberikan pengaruh yang kuat terhadap enzim tersebut. Dari 20 senyawa, *N Plus O Count* memiliki rata-rata tertinggi sedangkan senyawa dengan rata-rata terendah adalah *A Log P 98 Unknown.* Pada enzim *aofb* terdapat 26 senyawa yang mempunyai rata-rata dibawah nol dan 44 senyawa yang mempunyai rata-rata diatas nol. Senyawa-senyawa yang memiliki rata-rata dibawah nol dan merupakan senyawa yang memberikan pengaruh yang kuat mayoritas berasal dari tipe struktur spesifik.



Gambar 4.5 Nilai rata-rata variabel pada enzim *aofb*

Berikut ini adalah grafik yang menunjukkan sebaran rata-rata pada enzim *cah2*.

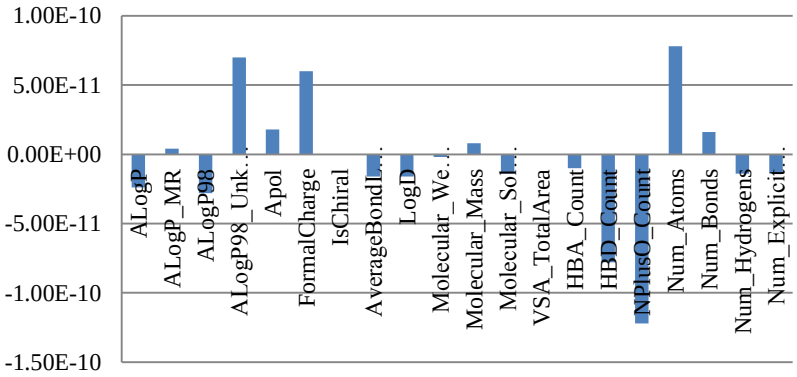


Gambar 4.6 Nilai rata-rata variabel pada enzim *cah2*

Berdasarkan Gambar 4.6 yang merupakan sebaran nilai rata-rata kandungan 20 senyawa penyusun enzim *cah2* diketahui bahwa terdapat rata-rata senyawa yang berada diatas nol dan berada dibawah nol. Senyawa-senyawa dengan kandungan senyawa bernilai dibawah nol cenderung memberikan pengaruh yang kuat terhadap enzim tersebut. Berdasarkan Gambar 4.6

senyawa dengan rata-rata tertinggi adalah *A Log P 98Unknown* sedangkan senyawa dengan rata-rata terendah adalah *Is Chiral.* Pada enzim *cah2* terdapat 28 senyawa yang mempunyai rata-rata dibawah nol dan 43 senyawa yang mempunyai rata-rata diatas nol. Senyawa-senyawa yang memiliki rata-rata dibawah nol dan merupakan senyawa yang memberikan pengaruh yang kuat mayoritas berasal dari tipe struktur spesifik.

Berikut ini adalah grafik yang menunjukkan sebaran rata-rata pada enzim *hs90a.*



Gambar 4.7 Nilai rata-rata variabel pada enzim *hs90a*

Sebaran nilai rata-rata kandungan 20 senyawa penyusun enzim *cah2* ditunjukkan pada Gambar 4.7. Terdapat rata-rata senyawa yang berada diatas nol dan berada dibawah nol. Senyawa-senyawa dengan kandungan senyawa bernilai dibawah nol cenderung memberikan pengaruh yang kuat terhadap enzim tersebut. Dari 20 senyawa, senyawa dengan rata-rata tertinggi adalah *Num Atom S* sedangkan senyawa dengan rata-rata terendah adalah *N Plus O Count.* Pada enzim *hs90a* terdapat 37 senyawa yang mempunyai rata-rata dibawah nol dan 32 senyawa yang mempunyai rata-rata diatas nol. Senyawa-senyawa yang memiliki rata-rata dibawah nol dan merupakan senyawa yang memberikan pengaruh yang kuat mayoritas berasal dari tipe struktur spesifik.

4.2 Analisis Regresi Logistik Biner untuk Senyawa Enzim Pada Database *DUD-E*

Analisis regresi logistik biner dilakukan terhadap ketiga jenis enzim, dimana enzim *aofb* terdiri dari 672 pengamatan dengan 70 variabel prediktor, enzim *cah2* terdiri dari 3340 pengamatan dengan 71 variabel prediktor dan enzim *hs90a* terdiri dari 500 pengamatan dengan 69 variabel prediktor. Masing-masing data tersebut dapat dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Pembagian data menjadi dua bagian bertujuan untuk mendapatkan suatu model yang optimal. Kombinasi data *training* dan data *testing* yang digunakan dalam penelitian ini adalah 90% :10% dan 85%:15%. Data *training* digunakan untuk mengestimasi parameter model dan data *testing* digunakan untuk mencerminkan kebaikan model dalam mengklasifikasikan data baru.

Analisis regresi logistik biner dilakukan dengan tahapan awal uji serentak, kemudian uji parsial hingga diperoleh suatu model. Berikut diberikan penjelasan untuk masing-masing tahapan analisis klasifikasi mengggunakan regresi logistik biner menggunakan kombinasi data *training* dan *testing* sebesar 90%:10%. Pada analisis ini pemilihan model terbaik dilakukan dengan menggunakan metode *forward* yaitu dengan cara memilih variabel yang mempunyai nilai R-*square* tertinggi.

4.2.1 Uji Serentak Terhadap Variabel-variabel Yang Berpengaruh Terhadap Senyawa Enzim Pada *Database DUD-E*

Langkah awal dalam analisis regresi logistik biner adalah uji serentak terhadap variabel-variabel yang berpengaruh terhadap masing-masing enzim pada database *DUD-E*. Pada analisis ini digunakan data *training* pada masing-masing enzim. Berikut ini adalah pengujian serentak pada ketiga jenis enzim.

Tabel 4.1 Uji Serentak Regresi Logistik Biner

Enzim	*-2 log likelihood*	*df*
aofb	673,2	12
cah2	3357,5	8
hs90a	507,2	4

Statistik uji G merupakan nilai *likelihood ratio test* yang mengikuti distribusi *chi-squared*. Pada enzim *aofb* didapatkan nilai G sebesar 673,2 yang menunjukkan nilai lebih besar dibandingkan dengan nilai $\chi 2_{(12;0,05)} = 18,54$ sehingga dapat disimpulkan bahwa variabel yang diguankan pada enzim *aofb* signifikan terhadap model. Pada enzim *cah2* didapatkan nilai G sebesar 3357,5 yang menunjukkan nilai lebih besar dibandingkan dengan nilai $\chi 2_{(8;0,05)} = 13,36$ sehingga dapat disimpulkan bahwa variabel yang digunakan pada enzim *cah2* signifikan terhadap model. Pada enzim *hs90a* didapatkan nilai G sebesar 507,2 yang menunjukkan nilai lebih besar dibandingkan dengan nilai $\chi 2_{(4;0,05)} = 7,77$ sehingga dapat disimpulkan bahwa variabel yang digunakan pada enzim *cah2* signifikan terhadap model.

4.2.2 Uji Parsial Terhadap Variabel-variabel Yang Berpengaruh Terhadap Enzim Pada *Database DUD-E*

Langkah kedua setelah dilakukan uji serentak, selanjutnya dilakukan uji parsial terhadap variabel-variabel yang berpengaruh terhadap jenis enzim pada *Database DUD-E*. Berdasarkan hasil analisis uji serentak didapatkan bahwa variabel yang digunakan signifikan terhadap model namun belum diketahui variabel mana pada masing-masing enzim yang berpengaruh secara signifikan terhadap enzim tersebut, sehingga dilakukan pengujian parsial untuk mengetahui variabel prediktor yang mempengaruhi masing-masing enzim. Pengujian parsial pada masing-masing enzim untuk mengetahui variabel yang berpengaruh terhadap enzim pada *database DUD-E* terdapat pada Tabel 4.2

Tabel 4.2 Uji Parsial Regresi Logistik Biner

Enzim	Variabel prediktor	B	*df*	*p-value*	*Exp(B)*
Aofb	Konstan	-2,462	1	0,000	
	ALog P98 Unknown	0,586	1	0,000	1,796
	Is Chiral	0,513	1	0,007	1,671
	Average Bond Length	-1,041	1	0,000	0,353
	Num_Aromatic Rings	-4,179	1	0,005	0,015

Tabel 4.2 Uji Parsial Regresi Logistik Biner (*Lanjutan*)

Enzim	Variabel prediktor	B	*df*	*p-value*	*Exp(B)*
aofb	*Num Chains*	-7,727	1	0,000	0,000
	Num ExplicitAtoms	5,584	1	0,000	266,03
	Num Aromatic Bonds	3,604	1	0,021	36,745
	Num H Acceptors	-1,454	1	0,000	0,234
	Num H Donors	-1,535	1	0,000	0,215
	Num H Donors Lipinski	-0,817	1	0,006	0,442
	Rad Of Gyration	1,649	1	0,000	5,204
cah2	konstan	-2,634	1	0,956	
	Molecular Mas	2,346	1	0,000	10,445
	HBA Count	1,369	1	0,000	3,93
	N Plus O Count	-2,032	1	0,000	0,131
	Num Rings6	1,316	1	0,000	3,729
	Num_Stereo Bonds	-1,116	1	0,000	0,328
	Num AtomClasses	-3,194	1	0,000	0,041
	Num_H_Acceptors	-2,913	1	0,996	0,054
	Molecular 3D Polar SASA	2,684	1	0,000	14,639
hs90a	konstan	-12,469	1	0,992	
	Is Chiral	-10,615	1	0,992	0,000
	Num_Rings 9 Plus	7,968	1	0,997	0,003
	Num_H Acceptors	1,903	1	0,000	6,706
	Rad Of Gyration	-1,981	1	0,000	0,138

Setelah dilakukan uji serentak, selanjutnya dilakukan uji parsial pada masing-masing variabel prediktor. Pada enzim *aofb* model terbaik didapatkan pada iterasi ke 12 dan terdapat 12 variabel prediktor yang masuk kedalam model, tiap-tiap variabel yang telah masuk kedalam model yaitu variabel *ALogP98 Unknown, IsChiral, LogD, HBD Count, Num Explicit Atoms, Num AromaticBonds, Num AromaticRings, Num Chains, Num H*

Acceptors, Num H Donors, Num H Donors Lipinski dan Rad Of Gyration signifikan ditunjukkan dengan *p-value* < α. Pada enzim *cah2* model terbaik didapatkan pada iterasi ke 8 dan terdapat 8 variabel prediktor yang masuk kedalam model yaitu variabel *Molecular Mas, HBA Count, N Plus O Count, Num Rings6, Num_Stereo Bonds, Num Atom Classes, Num_H_Acceptors* dan *Molecular 3D Polar SASA*. Variabel *Molecular Mas, HBA Count, N Plus O Count, Num Rings6, Num_Stereo Bonds, Num Atom Classes* dan *Molecular 3D Polar SASA* signifikan ditunjukkan dengan *p-value* < α namun terdapat satu variabel yang tidak signifikan yaitu variabel *Num_H_Acceptors* hal ini ditunjukkan dengan *p-value* > α. Pada enzim *hs90a* model terbaik didapatkan pada iterasi ke 4 dan terdapat 4 variabel prediktor yang masuk kedalam model yaitu variabel *Is Chiral, Num_Rings9Plus, Num H Acceptors* dan *Rad Of Gyration.* Variabel *Num H Acceptors* dan *Rad Of Gyration* signifikan ditunjukkan dengan *p-value* < α namun terdapat dua variabel yang tidak signifikan yaitu variabel *Is Chiral* dan *Num_Rings 9Plus* ditunjukkan dengan *p-value* > α.

Selanjutnya dari variabel yang signifikan akan dilakukan interpretasi model menggunakan nilai *odds ratio*. Pada enzim *aofb* terdapat 6 variabel yaitu *ALogP98 Unknown, IsChiral, HBD Count, Num ExplicitAtoms, Num Aromatic Bond* dan *Rad Of Gyration* yang mempunyai nilai *odd ratio* lebih besar dari 1, hal ini berarti bahwa 6 variabel tersebut mempunyai pengaruh positif terhadap model yang terbentuk. Pada senyawa *ALogP98 Unknown* didapatkan nilai *odd ratio* sebesar 1,796 yang berarti bahwa semakin tinggi kadar senyawa *ALogP98 Unknown* maka kecenderungan untuk diklasifikasikan kedalam inhibitor buruk adalah sebesar 1,796 kali dibandingkan dengan inhibitor baik. Pada senyawa *Is Chiral* didapatkan *odd ratio* sebesar 1,6 yang berarti bahwa semakin tinggi kandungan senyawa *Is Chiral* maka kecenderungan untuk diklasifikasikan kedalam inhibitor buruk adalah 1,6 kali dibandingkan dengan inhibitor baik. Pada senyawa *HBD Count odd ratio* sebesar 2,4 berarti bahwa semakin tinggi kadar senyawa *HBD Count* maka memiliki kecenderungan 2,4 kali untuk diklasifikasikan kedalam inhibitor buruk dari inhibitor baik. Pada senyawa *Num ExplicitAtoms* nilai *odd ratio* sebesar

266 berarti bahwa semakin tinggi kadar senyawa *Num Explicit Atoms* maka memiliki kecenderungan 266 kali untuk menjadi inhibitor buruk dibandingkan dengan inhibitor baik. Pada senyawa *Num Aromatic Bond odd ratio* sebesar 36,7 berarti bahwa semakin tinggi kandungan senyawa pada *Num Aromatic Bond* maka memiliki kecenderungan 36,7 kali untuk menjadi inhibitor buruk dibandingkan dengan menjadi inhibitor baik. Nilai *odd ratio* sebesar 5,2 pada senyawa *Rad Of Gyration* berarti bahwa semakin tinggi kadar senyawa *Rad Of Gyration* memiliki kecenderungan untuk menjadi inhibitor buruk sebesar 5,2 kali dibandingkan dengan inhibitor baik.

Pada enzim *cah2* terdapat 4 variabel yaitu *Molecular Mas, HBA count, Num Rings6* dan *Molecular 3D* yang mempunyai nilai *odd ratio* lebih besar dari 1, hal ini berarti bahwa 4 variabel tersebut mempunyai pengaruh positif terhadap model yang terbentuk. Nilai *odd ratio* sebesar 10,4 pada senyawa *Molecular Mas* memiliki arti bahwa semakin tinggi kadar senyawa *Molecular Mas* maka memiliki kecenderungan 10,4 kali untuk menjadi inhibitor buruk dibandingkan dengan inhibitor baik. Pada senyawa *HBA count* nilai *odd ratio* sebesar 3,93 menunjukkan bahwa semakin tinggi kadar senyawa *HBA count* maka memiliki kecenderungan 3,93 kali untuk menjadi inhibitor buruk dibandingkan dengan inhibitor baik. Pada senyawa *Num Rings6 odd ratio* sebesar 3,7 menunjukkan bahwa semakin tinggi kadar senyawa *Num Rings6* maka memiliki kecenderungan 3,7 kali untuk diklasifikasikan menjadi inhibitor buruk dibandingkan dengan inhibitor baik. Nilai *odd ratio* sebesar 14,6 pada senyawa *Molecular 3D* menunjukkan bahwa semakin tinggi kadar senyawa *Molecular 3D* maka memmiliki kecenderungan 14,6 kali untuk menjadi inhibitor buruk dibandingkan dengan inhibitor baik.

Pada enzim *hs90a* terdapat 1 variabel yaitu *Num_H Acceptors* yang mempunyai nilai *odd ratio* lebih besar dari 1 hal ini berarti bahwa variabel tersebut mempunyai pengaruh positif terhadap model yang terbentuk. Nilai *odd ratio* sebesar 6,7 pada senyawa *Num_H Acceptors* berarti bahwa semakin tinggi kandungan senyawa *Num_H Acceptors* maka memiliki

kecenderungan 6,7 kali untuk menjadi inhibitor buruk dibandingkan dengan inhibitor baik.

4.2.3 Model Regresi Logistik Biner dalam Kasus Klasifikasi Enzim Pada *Database DUD-E*

Tahapan selanjutnya, setelah dilakukan uji serentak dan uji parsial adalah membuat persamaan model regresi logistik biner yang digunakan untuk pengklasifikasian baik data *training* maupun data *testing* yang selanjutnya dilakukan interpretasi terhadap model tersebut. Berdasarkan nilai parameter pada analisis uji parsial yang ditampilkan pada Tabel 4.2 didapatkan persamaan logit untuk inhibitor baik (*Ligand*) pada enzim *aofb, cah2 dan hs90a* berturut-turut adalah sebagai berikut.

Transformasi Logit untuk enzim *aofb*

$$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -2{,}46 + 0{,}58x_4 + 0{,}51x_7 - 1{,}04x_8 + 0{,}88x_{15} + 5{,}58x_{21} + 3{,}6x_{29} - 4{,}18x_{32} - 7{,}7x_{41} - 1{,}5x_{51} - 1{,}5x_{32} - 0{,}82x_{54} + 1{,}65x_{67}$$

Transformasi Logit untuk enzim *cah2*

$$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -2{,}63 + 2{,}34x_{11} + 1{,}37x_{14} - 2{,}03x_{16} + 1{,}31x_{37} - 1{,}12x_{44} - 3{,}19x_{46} - 2{,}91x_{51} + 2{,}68x_{70}$$

Transformasi Logit untuk enzim *hs90a*

$$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -12{,}47 - 10{,}61x_7 + 7{,}37x_{39} + 1{,}9x_{50} + 1{,}98x_{66}$$

4.2.4 Uji kesesuaian model

Uji kesesuaian model dilakukan untuk mengetahui model regresi logistik biner yang terbentuk pada klasifikasi enzim yang akan digunakan dalam analisis pengklasifikasian sudah sesuai atau belum. Uji kesesuain model ditampilkan pada Tabel 4.3.

Tabel 4.3 Uji Kesesuaian Model

enzim	*chi-square*	*df*	*p-value*	*Nagelkerke R Square*
aofb	28,789	8	0,001	70,5
cah2	45,022	8	0,000	73,9
hs90a	1,748	8	0,988	83

Pada enzim *aofb* dan *cah2* didapatkan *p-value* sebesar 0,001 dan 0,000 yang menunjukan nilai kurang dari α=0,05 sehingga dapat disimpulkan bahwa dengan tingkat keyakinan sebesar 95% model yang didapatkan pada enzim *aofb* dan *cah2* tidak mampu menjelaskan data atau tidak sesuai. Pada enzim *hs90a* didapatkan *p-value* sebesar 0,988 yang menunjukkan nilai nilai lebih besar dari α=0,05 sehingga dapat disimpulkan bahwa pada tingkat keyakinan sebesar 95% model yang didapatkan pada enzim *hs90a* mampu menjelaskan data. *Nagelkerke R Square* mempunyai interpretasi mirip dengan koefisien determinasi pada regresi linier. Pada Tabel 4.3 didapatkan nilai *Nagelkerke R Square* untuk enzim *aofb* sebesar 70,5, enzim *cah2* sebesar 73,9 dan untuk enzim *hs90a* sebesar 83. Hal ini menunjukkan bahwa proporsi varians klasifikasi enzim *aofb* yang dapat dijelaskan oleh model adalah sebesar 70,5 persen, proporsi varians klasifikasi enzim *cah2* yang dapat dijelaskan oleh model adalah sebesar 73,9 persen dan proporsi varians klasifikasi enzim *hs90a* adalah sebesar 83 persen. Nilai-nilai ini hanya pendekatan saja, karena pada regresi logistik koefisien determinasi tidak dapat dihitung seperti regresi linier. Sehingga perlu lebih diperhatikan adalah seberapa banyak kita dapat memprediksi dengan benar yang tercermin dari nilai *Classification Plot.*

4.2.5 Klasifikasi Enzim Pada *Database DUD-E*

Klasifikasi enzim digunakan untuk mengetahui ketepatan model yang telah terbentuk dalam mengestimasi parameter model menggunakan data *training* dan untuk mengetahui ketepatan dalam mengklasifikasikan data baru menggunakan data *testing*. Ketepatan klasifikasi enzim menggunakan metode regresi logistik biner terdapat pada Tabel 4.4.

Berdasarkan Tabel 4.4 diketahui bahwa pada enzim *aofb* ketepatan klasifikasi untuk data *training* sebesar 90,4 persen sedangkan ketepatan klasifikasi untuk data *testing* sebesar 91,04 persen. Pada enzim *aofb,* model regresi logistik biner mempunyai ketepatan 90,4 persen dalam mengestimasi parameter model sedangkan model regresi logistik biner mempunyai ketepatan 91,04 persen dalam mengklasifikasikan data enzim baru. Ketepatan klasifikasi enzim *cah2* untuk data *training* sebesar

91,7 persen sedangkan ketepatan klasifikasi untuk data *testing* sebesar 91,3 persen. Hal tersebut berarti bahwa pada enzim *cah2* model regresi logistik biner mempunyai ketepatan 91,7 persen dalam mengestimasi parameter model sedangkan model regresi logistik biner mempunyai ketepatan 91,3 persen dalam mengklasifikasikan data enzim baru. Pada enzim *hs90a* ketepatan klasifikasi untuk data *training* sebesar 94 persen sedangkan ketepatan klasifikasi untuk data *testing* sebesar 100 persen. Hal tersebut berarti bahwa pada enzim *hs90a* model regresi logistik biner mempunyai ketepatan 94 persen dalam mengestimasi parameter model sedangkan model regresi logistik biner mempunyai ketepatan 100 persen dalam mengklasifikasikan data enzim baru.

Tabel 4.4 Klasifikasi Enzim

	Enzim	observasi	prediksi 0	prediksi 1	total	1-APER
Data *Training*	*Aofb*	0	436	21	457	90,4
		1	37	111	148	
	Cah	0	2169	96	2265	91,7
		1	154	587	741	
	hs90a	0	319	18	337	94
		1	9	104	113	
Data *Testing*	*Aofb*	0	46	5	51	91,04
		1	1	15	16	
	Cah	0	230	19	249	91,3
		1	10	75	85	
	hs90a	0	38	0	38	100
		1	0	12	12	

4.2.6 Pemilihan Kombinasi Data *Training* dan Data *Testing* Terbaik dalam Analisis Regresi Logistik Biner

Proses analisis regresi logistik biner yang telah dijelaskan sebelumnya dijalankan pula pada kombinasi data *training* dan *testing* lainnya yaitu 85%:15%. Berdasarkan hasil pengolahan analisis regresi logistik biner, maka diperoleh hasil ketepatan klasifikasi (1-APER) untuk masing-masing kombinasi data *training* dan *testing* sebagai berikut.

Tabel 4.5 Perbandingan *Total Accuracy Rate* Beberapa Kombinasi Data

Enzim	Kombinasi Data *Training* dan *Testing*	*Total Accuracy Rate* (1-APER) (dalam %)	
		Data *Training*	Data *Testing*
aofb	90%:10%*	90,4	91,04
	85%:15%	91,1	88,1
cah2	90%:10%*	91,7	91,3
	85%:15%	92,4	81,4
hs90a	90%:10%*	94	100
	85%:15%	93,6	77,3

*kombinasi data *training* dan *testing* terpilih

Tabel 4.5 adalah perbandingan ketepatan klasifikasi menggunakan kombinasi data *training testing* 90%:10% dan 85%:15%. Apabila pada data *training* memberikan nilai ketepatan klasifikasi yang paling tinggi akan tetapi pada data *testing* tidak memberikan nilai ketepatan klasifikasi yang paling tinggi begitu juga sebaliknya sehingga yang dipilih adalah kombinasi data *training* dan *testing* yang memberikan nilai ketepatan klasifikasi tertinggi pada ketepatan akurasi data *testing*.

Pada enzim *aofb* dengan proporsi data *training testing* 90%:10% diperoleh ketepatan klasifikasi untuk data *training* sebesar 90,4 persen dan ketepatan klasifikasi untuk data *testing* sebesar 91,04 persen. Sedangkan dengan proporsi data *training testing* 85%:15% diperoleh ketepatan klasifikasi untuk data *training* sebesar 99,1 dan ketepatan klasifikasi untuk data *testing* sebesar 88,1 persen.

Pada enzim *cah2* dengan proporsi data *training testing* 90%:10% diperoleh ketepatan klasifikasi untuk data *training* sebesar 91,7 persen dan ketepatan klasifikasi untuk data *testing* sebesar 91,3 persen. Sedangkan dengan proporsi data *training testing* 85%:15% diperoleh ketepatan klasifikasi untuk data *training* sebesar 92,4 dan ketepatan klasifikasi untuk data *testing* sebesar 81,4 persen.

Pada enzim *hs90a* dengan proporsi data *training testing* 90%:10% diperoleh ketepatan klasifikasi untuk data *training* sebesar 94 persen dan ketepatan klasifikasi untuk data *testing* sebesar 100 persen. Sedangkan dengan proporsi data *training*

testing 85%:15% diperoleh ketepatan klasifikasi untuk data *training* sebesar 93,6 dan ketepatan klasifikasi untuk data *testing* sebesar 77,3 persen.

Secara keseluruhan pada ketiga jenis enzim kombinasi data *training testing* 90%:10% memberikan ketepatan klasifikasi yang lebih tinggi pada data *testing* sedangkan kombinasi data *training testing* 85%:15% memberikan ketepatan klasifikasi yang tinggi untuk data *training* sehingga kombinasi data *training testing* yang terpilih sebagai kombinasi terbaik adalah kombinasi data *training testing* 90%:10% karena memiliki ketepatan klasifikasi yang tinggi pada data *testing* dimana data *testing* menunjukkan kemampuan model untuk mengklasifikasikan data baru.

4.3 Analisis *Logistic Regression Ensembles* (*Lorens*) Untuk Klasifikasi Enzim pada *Database DUD-E*

Klasifikasi menggunakan regresi logistik biner mempunyai kelemahan yaitu tidak dapat memberikan keseimbangan antara *sensitivity* dan *specitificity* pada data yang tidak mempunyai respon positif dan negatif yang tidak seimbang, metode *logistic regression ensembles* (*Lorens*) dapat menyeimbangkan *sensitivity* dan *specitificity* menggunakan *threshold* optimal.

Analisis *logistic regression ensembles* (*Lorens*) merupakan suatu metode *ensemble* yang dilakukan untuk mendapatkan ketepatan klasifikasi yang tinggi pada *high dimensional data.* Tahapan yang dilakukan dalam analisis *logistic regression ensembles* (*Lorens*) adalah menghitung nilai *treshold* optimal, menentukan jumlah partisi, menentukan jumlah *ensemble*s dan membentuk model dari partisi yang terbentuk dalam satu *ensemble*. Ketepatan klasifikasi yang didapatkan merupakan nilai mayoritas *voting* dari semua *ensemble* yang terbentuk. Pada penelitian ini analisis *logistic regression ensembles* (*Lorens*) dilakukan dengan membagi data menjadi data *training* dan data *testing* dengan proporsi 90%:10% dan 85%:15%. Model yang terbentuk berasal dari 90% dan 85% data *training* kemudian model tersebut dievaluasi menggunakan 10% dan 15% data *testing*. Jumlah partisi yang akan dicobakan dalam penelitian ini adalah 2,3,4,5,6,7,8,9 dan 10 serta akan digunakan jumlah *ensemble* sebanyak 10 karena berdasarkan penelitian sebelumnya

ketepatan klasifikasi yang tinggi akan didapatkan dengan jumlah *ensemble* tersebut. Berikut ini adalah tahapan analisis *Logistic regression ensembles* (*lorens*) pada ketiga jenis enzim yaitu *aofb*, *cah2* dan *hs90a* menggunakan kmbinasi data training testing 90%:10%.

4.3.1 Penentuan Nilai *Treshold*

Treshold yang biasa digunakan dalam klasifikasi untuk respon biner adalah 0.5. Namun akurasi klasfikasi tidak akan baik apabila proporsi kelas 1 dan 0 tidak seimbang, sehingga nilai *treshold* optimum dicari berdasarkan respon positif dan negatif. *Treshold* optimum nantinya akan digunakan sebagai nilai ambang batas keputusan untuk mengklasifikasikan data kedalam jenis inihibitor baik dan inhibitor buruk. Berikut ini adalah perhitungan *treshold* optimum pada ketiga jenis enzim.

Tabel 4.6 Nilai *Treshold*

Enzim	*Ligand*	*Decoy*	*Treshold*
aofb	148	457	0,3723
cah2	741	2265	0,3732
hs90a	113	337	0,3755

Perhitungan *treshold* dicari dengan mengguakan 90% data *testing* yaitu 605 data pada enzim *aofb*, 3006 data pada enzim *cah2* dan 450 data pada enzim *hs90a*. Pada enzim *aofb* terdapat 148 pengamatan dengan respon positif dan 457 pengamatan dengan respon negatif sehingga didapatkan *treshold* sebesar 0,3723. Pada enzim *cah2* terdapat 741 pengamatan dengan respon positif dan 2265 pengamatan dengan respon negatif sehingga didapatkan *treshold* sebesar 0,3732. Pada enzim *hs90a* terdapat 150 pengamatan dengan respon positif dan 337 pengamatan dengan respon negatif sehingga didapatkan *treshold* sebesar 0,3755. Nilai *treshold* tersebut selanjutnya akan digunakan sebagai nilai ambang batas untuk mengklasifikasikan data enzim pada masing-masing *ensemble*.

4.3.2 Random Partisi dan Pembentukan Model

Partisi variabel secara acak dilakukan untuk meminimalkan korelasi pada suatu *ensemble* dimana random partisi dipilih

dengan distribusi yang sama sehingga diasumsikan tidak terjadi bias diantara masing-masing partisi. Langkah-langkah random partisi, pembentukan model dan penentuan ketepatan klasifikasi pada ketiga enzim adalah sama. Pembagian variabel kedalam masing-masing sub ruang dilakukan secara random dan memiliki ukuran yang hampir sama antar masing-masing sub ruang. Tahapan random partisi dan pembentukan model dengan 10 *ensemble* pada enzim *aofb* dijelaskan pada Tabel 4.7

Tabel 4.7 Random Partisi

Partisi	variabel	*ensemble*									
		1	2	3	4	5	6	7	8	9	10
2	*A Log P*	2	1	1	1	2	1	1	1	1	2
	A Log P MR	1	2	1	1	1	1	1	2	1	2
	A Log P98	2	1	1	1	1	1	2	1	2	2
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	Molecular_3D_SAVol	2	2	2	1	2	2	2	2	1	2
3	*A Log P*	2	3	1	2	1	1	1	3	3	2
	A Log P MR	2	3	3	3	3	3	1	3	1	3
	A Log P98	1	3	1	3	1	3	3	2	2	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	Molecular_3D_SAVol	3	3	1	2	3	2	1	2	3	3
4	*A Log P*	3	4	4	3	2	4	4	1	2	1
	A Log P MR	1	3	3	3	4	1	4	4	3	3
	A Log P98	3	2	4	4	4	3	1	2	2	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	Molecular_3D_SAVol	1	1	2	2	2	1	4	2	2	3
10	*A Log P*	3	9	9	2	8	2	6	3	9	1
	A Log P MR	9	3	10	2	4	3	3	4	9	4
	A Log P98	1	7	5	6	7	3	10	10	3	5
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	Molecular_3D_SAVol	1	4	6	6	4	8	6	9	7	5

Dengan menggunakan jumlah partisi sebanyak 2 maka 70 variabel prediktor pada enzim *aofb* akan dialokasikan kedalam 2 sub ruang, yaitu partisi 1 dan partisi 2. Pada masing-masing-masing *ensemble* akan terbentuk 2 model sehingga akan terbentuk 20 model untuk 10 *ensemble* yang terbentuk. Pada *ensemble*

pertama variabel *A Log P* dialokasikan kedalam partisi 2, variabel *A Log P MR* dialokasikan kedalam partisi 1, variabel *A Log P 98* dialokasikan kedalam partisi 2 sampai dengan variabel *Molecular 3D SAVol* dialokasikan kedalam partisi 2. Pengalokasian yang sama juga dilakukan pada *ensemble* kedua, ketiga dan seterusnya.

Dengan menggunakan partisi sebanyak 3, maka 70 variabel prediktor pada enzim *aofb* akan dialokasikan kedalam 3 sub ruang yaitu partisi 1, partisi 2 dan partisi 3. Pada masing-masing *ensemble* akan terbentuk 3 model sehingga akan terbentuk 30 model untuk 10 *ensemble*. Dengan mempartisi variabel kedalam 4 sub ruang maka 70 variabel pada enzim *aofb* akan dialokasikan kedalam 4 sub ruang yaitu partisi 1, partisi 2, partisi 3 dan partisi 4 sehingga akan terbentuk 40 model untuk 10 *ensemble.* Pengalokasian variabel yang sama berlaku untuk partisi yang lain yaitu partisi 5,6,7,8,9 dan 10. Setelah masing-masing variabel dialokasikan sesuai dengan partisi maka akan terbentuk model regresi logistik dari masing-masing partisi pada suatu *ensemble.* Misalkan digunakan partisi sebanyak 2 maka model regresi logistik yang terbentuk pada 10 *ensemble* ditampilkan pada Tabel 4.8

Tabel 4.8 Model regresi Logistik

Ens	Model Regresi Logistik Partisi 1	Model Regresi Logistik Partisi 2
1	$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -2{,}8 + 2{,}47x_1 + 1{,}2x_3 + 0{,}47x_4 + \cdots - 0{,}799x_{69}$	$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -3{,}9 + 9{,}57x_2 - 1{,}16x_5 + 302{,}4x_6 + \cdots + 2{,}26x_{70}$
2	$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -3{,}7 - 1{,}06x_1 + 2{,}45x_3 + 1{,}05x_6 + \cdots - 1{,}82x_{68}$	$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -3{,}15 - 3{,}56x_2 + 0{,}75x_4 + 0{,}04x_5 + \cdots + 3{,}82x_{70}$
3	$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -3{,}7 + 0{,}9x_4 + 346{,}6x_6 + 0{,}28x_9 + \cdots + 1{,}2x_{69}$	$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -2{,}9 - 9{,}81x_1 + 2{,}32x_2 + 1{,}5x_3 + \cdots + 3{,}35x_{70}$
⋮		
10	$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -2{,}95 - 0{,}89x_1 + 2{,}27x_2 + 1{,}97x_3 + \cdots + 14{,}7x_{68}$	$g(x) = ln\left(\frac{\pi(x)}{1-\pi(x)}\right) = -2{,}95 + 0{,}83x_4 + 0{,}83x_5 + \cdots + 0{,}64x_{69}$

Tabel 4.8 menunjukkan model yang terbentuk dari pengalokasian variabel prediktor kedalam 2 *sub ruang* yaitu partisi 1 dan partisi 2. Pada partisi 1 dan partisi 2 model melibatkan 35 variabel prediktor. Dengan mempartisi variabel prediktor menjadi 2 sub ruang maka akan terbentuk 20 model. Pembentukan model dengan cara pengalokasian variabel prediktor juga beraku untuk jumlah partisi yang lain. Setelah terbentuk 20 model dari partisi sebanyak 2, 30 model dari partisi sebanyak 3, 40 model dari partisi sebantak 4, 50 model dari partisi sebanyak 5, 60 model dari partisi sebanyak 6, 70 model dari partisi sebanyak 7, 80 model dari partisi sebanyak 8, 90 model dari partisi sebanyak 9 dan 100 model dari partisi sebanyak 10 selanjutnya model regresi logistik yang terbentuk akan digunakan untuk mengklasifikasikan data.

4.3.3 Ketepatan Klasifikasi

Langkah selanjutnya setelah model pada masing-masing partisi telah terbentuk maka akan dilakukan klasifikasi dengan menggunakan 90% data *training* dan 10% data *testing.* Langkah klasifikasi dilakukan dengan cara mensubtitusikan data *training* dan data *testing* terhadap model yang terbentuk pada setiap partisi dalam satu *ensemble*. Hasil subtitusi data *training* dan data *testing* pada masing-masing model yang terbentuk akan kemudian di rata-rata dan akan menghasilkan sebuah nilai tertentu yang disebut dengan probabilitas. Nilai probabilitas yang didapatkan dari satu *ensemble* yang terbentuk akan dibandingkan dengan nilai *threshold* optimal. Ketika nilai probabilitas melebihi *threshold* optimal maka data diklasifikasikan kedalam jenis inhibitor baik (*ligand*) dan ketika nilai probabilitas kurang dari *threshold* optimum maka data diklasifikasikan kedalam jenis inhibitor buruk (*decoy*).

Tahapan yang sama juga dilakukan pada *ensemble* kedua, ketiga dan seterusnya sampai dengan *ensemble* yang kesepuluh sehingga didapatkan klasifikasi probabilitas untuk sepuluh *ensemble* yang terbentuk. Klasifikasi akhir didapatkan dengan cara melakukan mayoritas *voting* pada masing-masing pengamatan dari sepuluh *ensemble* yang terbentuk. Apabila mayoritas dari sepuluh *ensemble* mengklasifikasikan pengamatan

kedalam jenis inhibitor baik (*ligand*) maka klasifikasi akhir diputuskan untuk mengklasifikasikan pengamatan kedalam jenis jenis inhibitor baik (*ligand*). Sebaliknya, apabila mayoritas dari sepuluh *ensemble* mengklasifikasikan pengamatan kedalam jenis inhibitor buruk (*decoy*) maka klasifikasi akhir diputuskan untuk mengklasifikasikan pengamatan kedalam jenis jenis inhibitor buruk (*decoy*). Berikut ini ketepatan klasifikasi pada enzim *aofb*.

Tabel 4.9 Ketepatan Klasifikasi Enzim *aofb*

partisi	*Training*			*Testing*		
	Sensitivity	*Specificity*	Akurasi	*Sensitivity*	*Specificity*	Akurasi
2	79,74	95,07	91,07	84,21	91,67	89,55
3	76,13	93,33	88,9	83,33	89,80	88,06
4	76,35	92,34	88,4	77,78	87,76	85,07
5	74,34	92,27	87,76	78,95	89,58	86,57
6	71,62	90,81	86,11	77,78	87,76	85,07
7	71,23	90,41	85,78	88,24	90,00	89,55
8	70,92	89,65	85,28	83,33	89,80	88,06
9*	70,71	89,46	85,12	88,24	90,00	89,55
10	72,79	89,55	85,78	82,35	88,00	86,57

**Partisi optimal yang terpilih*

Tabel 4.9 menunjukkan ketepatan klasifikasi dengan berbagai nilai partisi pada enzim *aofb*. Pada enzim *aofb sensitivity*, *specificity* dan akurasi semakin menurun seiring dengan bertambahnya partisi baik pada data *training* dan data *testing*. *Sensitivity* merupakan ketepatan untuk memprediksi respon positif, *specificity* merupakan ketepatan untuk memprediksi respon negatif dan akurasi merupakan ketepatan untuk memprediksi respon positif dan negatif. Dalam analisis *Lorens* pemilihan ukuran partisi terbaik tidak dilandaskan pada partisi yang memberikan ketepatan akurasi terbaik karena analisis *Lorens* untuk data klasifikasi enzim dalam kasus ini memberikan hasil terbaik untuk ukuran partisi yang semakin kecil dan cenderung memberikan kesimpulan bahwa adanya partisi dalam metode *Lorens* tidak dapat memberikan ketepatan klasifikasi yang tinggi. Oleh karena itu ukuran partisi terbaik dilihat dari penambahan jumlah variabel dalam partisi yang memberikan kontribusi kenaikan ketepatan klasifikasi yang cukup signifikan.

Pemilihan ketepatan klasifikai yang optimal didasarkan pada ketepatan klasifikasi pada data *testing* 10% karena ketepatan tersebut merupakan kemampuan model untuk mengklasifikasikan data baru. Partisi optimal yang terbentuk akan digunakan sebagai perbandingan dengan metode logistik regresi biner

Pada enzim *aofb* didapatkan ketepatan klasifikasi 10% data *testing* dengan partisi sebanyak 2 sebesar 89,55 persen, 88,05 persen untuk partisi sebanyak 3, 88,07 persen untuk partisi sebanyak 4, 86,56 persen untuk partisi sebanyak 5, 85,07 persen untuk partisi sebanyak 7, 88,05 persen untuk partisi sebanyak 8, 89,55 persen untuk partisi sebanyak 9 dan 86,56 persen untuk partisi sebanyak 10. Sehingga jumlah partisi optimum yang akan digunakan pada enzim *aofb* adalah partisi sebanyak 9 karena dengan mempartisi variabel kedalam 9 sub ruang didapatkan selisih ketepatan klasifikasi yang signifikan dari partisi 9 ke partisi 10 yaitu sebesar 2,99 persen yang berarti bahwa ketika partisi ditambahkan dari 9 ke 10 maka akan terjadi penurunan ketepatan klasifikasi sebesar 2,99 persen. Sehingga partisi optimal untuk enzim *aofb* adalah membagi variabel kedalam 9 sub ruang. Pengalokasian variabel prediktor menjadi 9 *sub ruang* mampu mengklasifikasikan data baru dengan ketepatan 88,95 persen. Berikut tabel ketepatan klasifikasi untuk enzim *cah2.*

Tabel 4.10 Ketepatan Klasifikasi Enzim *cah2*

partisi	*Training*			*Testing*		
	Sensitivity	*Specificity*	Akurasi	*Sensitivity*	*Specificity*	Akurasi
2	90,54	98,52	96,44	94,44	96,31	95,8
3	88,69	97,36	95,14	89,25	95,44	93,71
4*	86,98	96,74	94,24	89,77	93,90	92,81
5	84,89	95,31	92,71	87,65	90,91	90,12
6	83,72	92,88	90,78	88,00	89,19	88,92
7	82,00	91,51	89,42	85,53	88,76	88,02
8	84,62	90,24	89,12	92,19	87,04	88,02
9	83,68	89,52	88,38	89,55	87,27	87,72
10	82,90	88,05	87,12	91,53	85,45	86,52

**Partisi optimal yang terpilih*

S*ensitivity*, *specificity* dan akurasi untuk enzim *cah2* baik untuk data *training* dan data *testing* ditampilkan pada Tabel 4.10. Seperti halnya pada enzim *aofb*, pada enzim *cah2* nilai *sensitivity*, *specificity* dan akurasi cenderung menurun seiring dengan bertambahnya jumlah partisi. Partisi optimal juga akan dipilih berdasarkan penambahan jumlah variabel dalam partisi yang memberikan kontribusi kenaikan ketepatan klasifikasi yang cukup signifikan.

Akurasi 10% data *testing* yang didapatkan dengan partisi 2 sebesar 95,8 persen, dengan partisi 3 sebesar 93,71 persen, dengan partisi 4 sebesar 92,81 persen, dengan partisi 5 sebesar 90,12, dengan partisi 6 sebesar 88,92 persen, dengan partisi 7 dan 8 sebesar 88,02 persen, dengan partisi 9 sebesar 87,72 persen dan dengan partisi 10 sebesar 86,52 persen. Berdasarkan uraian tersebut partisi optimal yang terpilih adalah partisi variabel kedalam 4 sub ruang dengan akurasi sebesar 92,81 persen. Ketika jumlah partisi ditambah menjadi 5 maka akurasi akan turun 1,53 persen, penurunan akurasi tersebut cukup signifikan dibandingkan dengan penurunan akurasi dengan jumlah partisi lainnya, sehingga diputuskan bahwa partisi optimum pada enzim *cah2* adalah 4. Berikut tabel ketepatan klasifikasi untuk enzim *hs90a.*

Tabel 4.11 Ketepatan Klasifikasi Enzim *hs90a*

partisi	*Training*			*Testing*		
	Sensitivity	*Specificity*	Akurasi	*Sensitivity*	*Specificity*	Akurasi
2	100	100	100	80	100	94
3	95,76	100	98,88	100	100	100
4	90,98	99,39	97,11	100	100	100
5*	88,10	99,38	96,22	100	100	100
6	86,05	99,38	95,55	92,31	100	98
7	84,80	97,85	94,22	92,31	100	98
8	85,48	97,85	94,44	100	100	100
9	83,20	97,23	93,33	92,31	100	98
10	82,54	97,22	93,11	92,31	100	98

**Partisi optimal yang terpilih*

Ketepatan klasifikasi untuk enzim *hs90a* baik nilai *sensitivity*, *specificity* dan akurasi untuk berbagai nilai partisi untuk data *training* dan data *testing* ditabelkan pada Tabel 4.11. Berdasarkan Tabel 4.11, *sensitivity, specificity* dan akurasi pada data *training* terus menurun seiring dengan bertambahnya jumlah partisi. Pada data *testing* nilai *sensitivity*, *specificity* dan akurasi tidak selalu menurun seiring dengan penambahan jumlah partisi. Seperti halnya pada enzim *aofb* dan *cah2,* dari berbagai nilai partisi akan dipilih satu partisi optmal yaitu dengan cara memilih penambahan jumlah variabel dalam partisi yang memberikan kontribusi kenaikan ketepatan klasifikasi yang cukup signifikan.

Pada enzim *hs90a* didapatkan akurasi dengan mempartisi variabel kedalam 2 sub ruang sebesar 94 persen, 3,4,5 dan 8 sub ruang sebesar 100 persen, 6,7,9 dan 10 sub ruang sebesar 98 persen. Partisi optimal yang dipilih untuk enzim *hs90a* adalah 5 karena jika partisi ditambahkan menjadi 6 atau dalam kata lain variabel dikurangi maka akurasi akan menurun sebesar 2 persen. Pengalokasian variabel prediktor menjadi 5 sub ruang menghasilkan akurasi sebesar 100 persen. Selanjutnya akurasi dari jumlah partisi optimum pada masing-masing enzim yaitu *aofb* sebanyak 9 partisi, *cah2* sebanyak 4 partisi dan *hs90a* sebanyak 5 partisi akan dibandingkan dengan akurasi yang didapatkan pada analisis regresi logistik biner.

4.3.4 Pemilihan Kombinasi Data *Training* dan Data *Testing* Terbaik dalam Analisis *Logistic Regression Ensemble* (*Lorens*)

Proses analisis *logistic regression ensemble* (*Lorens*) yang telah dijelaskan sebelumnya dijalankan pula pada kombinasi data *training* dan *testing* lainnya yaitu 85%:15%. Berdasarkan hasil pengolahan analisis *logistic regression ensemble* (*Lorens*), maka diperoleh perbandingan akurasi untuk masing-masing kombinasi data *training* dan *testing* dengan menggunakan 9 partisi pada enzim *aofb*, 4 partisi pada enzim *cah2* dan 5 partisi pada enzim *hs90a*. Berikut ini adalah perbandingan akurasi dari kombinasi data *training testing* dengan kombinasi 90%:10% dan 85%:15%.

Tabel 4.12 Perbandingan *Total Accuracy Rate* Beberapa Kombinasi Data

Enzim	Kombinasi Data *Training* dan *Testing*	*Total Accuracy Rate* (1-APER) (dalam %) Data *Training*	Data *Testing*
aofb	90%:10%*	85,12	89,55
	85%:15%	86	85,14
cah2	90%:10%*	94,24	92,81
	85%:15%	94,57	92,6
hs90a	90%:10%*	96,22	100
	85%:15%	96,23	98,7

*kombinasi data *training* dan *testing* terpilih

Pada enzim *aofb* akurasi tertinggi pada data *training* yaitu sebesar 86 persen diperoleh pada kombinasi data *training* dan *testing* 85%:15% namun pada data *testing* akurasi tertinggi sebesar 89,55 persen diperoleh dari kombinasi data *training* dan *testing* 90%:10%. Pada enzim *cah2* akurasi tertinggi pada data *training* yaitu sebesar 94,57 persen diperoleh pada kombinasi data *training* dan *testing* 85%:15% namun pada data *testing* akurasi tertinggi sebesar 92,81 persen diperoleh dari kombinasi data *training* dan *testing* 90%:10%. Pada enzim *hs90a* akurasi tertinggi pada data *training* yaitu sebesar 96,23 persen diperoleh pada kombinasi data *training* dan *testing* 85%:15% namun pada data *testing* akurasi tertinggi sebesar 100 persen diperoleh dari kombinasi data *training* dan *testing* 90%:10%.

Berdasarkan uraian tersebut maka kombinasi data *training testing* terbaik pada analisis *logistic regression ensemble* (*Lorens*) adalah 90%:10% karena karena memiliki ketepatan klasifikasi yang tinggi pada data *testing* dimana data *testing* menunjukkan kemampuan model untuk mengklasifikasikan data baru.

4.3.5 Ketepatan Klasifikasi *Cross Validation*

Ketepatan klasifikasi pada metode *Lorens* dapat dihitung menggunakan metode evaluasi *Cross Validation* yang diharapkan dapat meningkatkan performa ketepatan klasifikasi. Pada evaluasi *Cross Validation* data juga dicobakan kedalam jumlah partisi yang berbeda yaitu 2,3,4,5,6,7,8,9 dan 10. Dalam penelitian ini akan digunakan10 *ensemble* dan 10 *fold Cross*

Validation yang berarti bahwa akan terbentuk 10 *fold* dimana masing-masing *fold* akan terdiri dari 10 model. Dengan menggunakan 2 partisi maka akan terbentuk 200 model, dengan menggunakan 3 partisi maka akan terbentuk 300 model dan seterusnya sampai dengan partisi 10 maka akan terbentuk 1000 model. Pada metode *Lorens* dengan evaluasi *Cross Validation* pertama-tama data dibagi menjadi 10 bagian. Sepersepuluh bagian data pertama akan digunakan sebagai data *testing* pada *fold* ke-1 dan sisanya akan digunakan sebagai data *training.* Sepersepuluh bagian data kedua akan digunakan sebagai data *testing* pada *fold* ke-2 dan sisanya digunakan sebagai data *training* dan seteerusnya sampai dengan sepersepuluh data kesepuluh akan digunakan sebagai data *testing* pada *fold* ke-10 dan sisanya digunakan sebagai data *training.* Seperti halnya pada metode *Lorens* pengambilan keputusan didasarkan pada *threshold* optimal. Pada masing-masing *fold* akan terbentuk 1 nilai *threshold* optimal sehingga akan terdapat 10 nilai *threshold.* Tabel 4.13 merupakan contoh *treshold* optimal yang terbentuk pada enzim *aofb* ketika data dipartisi kedalam 2 *sub ruang*.

Tabel 4.13 *Treshold* Optimal Evaluasi *Cross Validation*

fold	*Treshold* Optimal	*fold*	*Treshold* Optimal
1	0,375	6	0,376
2	0,375	7	0,371
3	0,376	8	0,378
4	0,376	9	0,375
5	0,375	10	0,373

Dengan menggunakan partisi sebanyak 2 pada *fold* pertama didapatkan *threshold* sebesar 0,375, yang menunjukkan bahwa pengamatan dengan nilai probabilitas yang lebih besar dari 0,375 akan diklasifikasikan kedalam jenis inhibitor baik (*ligand*) dan pengamatan dengan nilai probabilitas yang kurang dari 0,375 akan diklasifikasikan kedalam jenis inhibitor buruk (*decoy*).

Seperti halnya pada metode *Lorens* nilai probabilitas yang didapatkan merupakan hasil subtitusi sepersepuluh bagian data *testing* kedalam model yang terbentuk pada masing-masing

partisi. Nilai probabilitas pada masing-masing *ensemble* kemudian di *voting*. Apabila mayoritas dari sepuluh *ensemble* mengklasifikasikan pengamatan kedalam jenis inhibitor baik (*ligand*) maka klasifikasi akhir diputuskan untuk mengklasifikasikan pengamatan kedalam jenis jenis inhibitor baik (*ligand*). Sebaliknya, apabila mayoritas dari sepuluh *ensemble* mengklasifikasikan pengamatan kedalam jenis inhibitor buruk (*decoy*) maka klasifikasi akhir diputuskan untuk mengklasifikasikan pengamatan kedalam jenis jenis inhibitor buruk (*decoy*). Setelah model pada masing-masing *fold* telah terbentuk dan disubtitusikan dengan data *training* sehingga terbentuk ketepatan klasifikasi pada masing-masing *fold*. Selanjutnya ketepatan klasifikasi pada masing-masing *fold* digabungkan dan didapatkan ketepatan klasifikasi pada masing-masing enzim. Ketepatan Klasifikasi Evaluasi *Cross Validation* Pada Enzim *aofb* dengan berbagai nilai partisi tertera pada Tabel 4.14.

Tabel 4.14 Ketepatan Klasifikasi Evaluasi *Cross Validation* Pada Enzim *aofb*

Partisi	*Sensitivity*	*Specificity*	Akurasi
2	73,89	92,89	87,8
3	73,71	92,15	87,35
4*	72,83	91,58	86,76
5	71,1	90,98	85,86
6	71,18	90,64	85,71
7	70,3	89,74	84,97
8	69,64	89,74	84,82
9	70	89,06	84,52
10	70,89	89,11	84,82

**Partisi optimal yang terpilih*

Berdasarkan Tabel 4.14 dapat diketahui bahwa setiap penambahan partisi akan menurunkan nilai *sensitivity*, *specificity* dan akurasi. Ukuran partisi optimal yang akan dipilih pada evaluasi *cross validation* adalah penambahan jumlah variabel dalam partisi yang memberikan kontribusi kenaikan ketepatan klasifikasi yang cukup signifikan.

Pada enzim *aofb* di dapatkan akurasi sebesar 87,8 persen dengan mempartisi variabel kedalam 2 sub ruang, 87,35 persen untuk partisi kedalam 3 sub ruang, 86,75 persen untuk partisi kedalam 4 sub ruang, 85,86 persen untuk partisi kedalam 5 sub ruang, 85,71 persen untuk partisi kedalam 6 sub ruang, 84,97 persen untuk partisi kedalam kedalam 7 sub ruang, 84,52 persen untuk partisi kedalam 8 sub ruang, 84,52 persen untuk partisi kedalam 9 sub ruang dan 84,8 persen untuk partisi kedalam 10 sub ruang. Partisi optimal yang terpilih pada enzim *aofb* adalah dengan mempartisi variabel kedalam 4 sub ruang karena didapatkan selisih akurasi yang tinggi ketika jumlah partisi ditingkatkan kedalam 5 sub ruang dibandingkan dengan penambahan partisi yang lain yaitu sebesar 0,9 persen. Hal ini berarti bahwa ketika jumlah partisi ditambahkan dari 4 sub ruang menjadi 5 sub ruang maka akurasi akan menurun 0,9 persen. Ketepatan klasifikasi untuk enzim *cah2* terdapat pada Tabel 4.15

Tabel 4.15 Ketepatan Klasifikasi Evaluasi *Cross Validation* Pada Enzim *cah2*

Partisi	*Sensitivity*	*Specificity*	Akurasi
2	88,33	98,36	95,66
3	87,73	97,44	94,88
4	85,40	96,51	93,59
5*	84,34	94,62	92,07
6	81,99	92,45	90,00
7	81,09	90,83	88,68
8	81,10	89,69	87,90
9	82,81	88,70	87,57
10	82,55	88,09	87,07

**Partisi optimal yang terpilih*

S*ensitivity*, *specificity* dan akurasi untuk enzim *cah2* dengan berbagai nilai partisi terdapat pada Tabel 4.15. Tabel 4.15 menunjukkan bahwa nilai *sensitivity*, *specificity* dan akurasi pada enzim *cah2* terus menurun seiring dengan bertambahnya jumlah partisi. Dari berbagai nilai partisi tersebut akan dipilih satu partisi optimal dengan cara yang sama seperti pada analisis *Lorens* tanpa meggunakan evaluasi *cross validation* yaitu dengan cara memilih

jumlah partisi yang memiliki kenaikan akurasi yang signifikan ketika jumlah partisi ditambahkan.

Pada enzim *cah2* didapatkan akurasi sebesar 95,7 persen dengan partisi sebanyak 2, 94,8 persen dengan partisi sebanyak 3, 93,5 persen dengan partisi sebanyak 4, 92,1 persen dengan partisi sebanyak 5, 90 persen dengan partisi sebanyak 6, 88,6 persen dengan partisi sebanyak 7, 87,9 persen dengan partisi sebanyak 8, 87,5 persen dengan partisi sebanyak 9 dan 87,1 persen dengan partisi sebanyak 10. Partisi optimal yang terpilih pada enzim *cah2* adalah dengan mempartisi variabel kedalam 5 sub ruang karena didapatkan selisih akurasi cukup signiifkan ketika jumlah partisi ditingkatkan dari 5 sub ruang menjadi 6 sub ruang dibandingkan dengan penambahan partisi yang lain yaitu sebesar 2,1 persen. Pengalokasian variabel pada enzim *cah2* kedalam 5 sub ruang menghasilkan akurasi sebesar 92,07 persen. Berikut ini ketepatan klasifikasi pada enzim *hs90a* menggunakan evaluasi *cross validation.*

Tabel 4.16 Ketepatan Klasifikasi Evaluasi *Cross Validation* Pada Enzim *hs90a*

Partisi	*Sensitivity*	*Specificity*	Akurasi
2	88,65	100	96,80
3	84,78	97,79	94,20
4*	89,86	99,72	97,00
5	85,92	99,16	95,40
6	86,96	98,62	95,40
7	85,40	97,80	94,40
8	84,06	97,51	93,80
9	82,01	96,95	92,80
10	82,01	96,95	92,80

**Partisi optimal yang terpilih*

Analisis *Lorens* menggunakan evaluasi *cross validation* pada enzim *hs90a* menghasilkan nilai *sensitivity*, *specificity* dan akurasi seperti yang terdapat pada Tabel 4.16. Tidak berbeda dengan enzim yang lain nilai *sensitivity*, *specificity* dan akurasi pada enzim *hs90a* juga terus menurun seiring dengan bertambahnya jumlah partisi. Seperti pada bahasan sebelumnya

partisi optimal akan dipilih berdasarkan penambahan jumlah variabel dalam partisi yang memberikan kenaikan ketepatan klasifikasi yang cukup signifikan.

Pada enzim *hs90a*, dengan partisi 2 didapatkan akurasi sebesar 96,8 persen, dengan partisi 3 didapatkan akurasi sebesar 94,2 persen, dengan partisi 4 didapatkan akurasi sebesar 97 persen, dengan partisi 5 dan 6 didapatkan akurasi sebesar 95,4 persen, dengan partisi 7 didapatkan akurasi sebesar 94,4 persen, dengan partisi 8 didapatkan akurasi sebesar 93,8 persen, dengan partisi 9 didapatkan akurasi sebesar 92,8 persen dan dengan partisi 10 didapatkan akurasi sebesar 92,8 persen. Berdasarkan berbagai akurasi tersebut partisi optimal yang terpilih pada enzim *hs90a* adalah dengan mempartisi variabel kedalam 4 sub ruang karena didapatkan selisih akurasi yang tinggi ketika jumlah partisi ditingkatkan kedalam 5 sub ruang dibandingkan dengan penambahan partisi yang lain yaitu sebesar 1,6 persen. Hal ini berarti bahwa ketika jumlah partisi ditambahkan dari 4 sub ruang menjadi 5 sub ruang maka ketepatan klasifikasi akan menurun 1,6 persen. Sehingga diputuskan bahwa partisi optimal adalah mempartisi variabel kedalam 4 sub ruang dengan akurasi sebesar 97 persen.

4.4 Perbandingan Hasil Klasifikasi Regresi Logistik Biner dan *Logistic Regression Ensemble* (*Lorens*)

Metode yang digunakan dalam klasifikasi data enzim pada *database DUD-E* adalah regresi logistik biner dan *logistic regression ensemble*. Kriteria yang digunakan untuk membandingkan antara kedua metode tersebut *total accuracy rate* (1-APER) dari masing-masing data *training* dan data *testing* yang terpilih sebagai kombinasi data *training testing* terbaik. Metode klasifikasi yang terbaik dipilih sebagai metode pengklasifikasian data enzim pada *database DUD-E*. Berdasarkan analisis menggunakan metode *logistic regression ensemble* (*Lorens*) didapatkan partisi optimal untuk enzim *aofb* dengan membagi variabel kedalam 9 sub ruang, untuk enzim *cah2* dengan membagi variabel kedalam 4 sub ruang dan untuk enzim *hs90a* dengan membagi variabel kedalam 5 sub ruang. Berikut ini

adalah perbandingan akurasi dari regresi logistik biner dan *logistic regression ensemble* (*Lorens*).

Tabel 4.17 Perbandingan Klasifikasi Regresi Logistik Biner dan *Logistic Regression Ensemble* (*Lorens*)

Metode	enzim	ketepatan klasifikasi(%) data *training*	ketepatan klasifikasi(%) data *testing*
Regresi Logistik Biner	*aofb*	90,4	91,04
	cah2	91,7	91,3
	hs90a	94	100
Logistic Regression Ensemble	*aofb*	85,12	89,55
	cah2	94,24	92,81
	hs90a	96,22	100

Berdasarkan Tabel 4.17, analisis menggunakan regresi logistik biner menghasilkan ketepatan klasifikasi pada enzim *aofb* sebesar 90,4 persen untuk data *training* dan 91,04 persen untuk data *testing*. Pada enzim *cah2* menghasilkan ketepatan klasifikasi 91,7 persen untuk data *training* dan 91,3 persen untuk data *testing*. Pada enzim *hs90a* dihasilkan ketepatan klasifikasi 94 persen untuk data *training* dan 100 persen untuk data *testing*.

Klasifikasi menggunakan metode *logistic regression ensembles* (*Lorens*) dengan mempartisi variabel kedalam 9 sub ruang menghasilkan ketepatan klasifikasi pada enzim *aofb* 85,12 persen untuk data *training* dan 89,55 persen untuk data *testing*. Pada enzim *cah2* dengan mempartisi variabel kedalam 4 sub ruang didapatkan ketepatan klasifikasi sebesar 94,24 persen untuk data *training* dan 92,81 persen untuk data *testing*. Pada enzim *hs90a* dengan mempartisi variabel kedalam 5 sub ruang didapatkan ketepatan klasifikasi sebesar 96,22 persen untuk data *training* dan 100 persen untuk data *testing*.

Pada enzim *aofb* ketepatan klasifikasi menggunakan metode regresi logistik biner lebih tinggi dibandingkan dengan ketepatan klasifikasi dengan metode *logistic regression ensemble* baik dengan data *training* maupun data *testing*. Pada enzim *cah2* ketepatan klasifikasi pada data *training* dan *testing* yang didapatkan dengan metode *logistic regression ensembles* (*Lorens*)

lebih tinggi dibandingkan dengan metode regresi logistik biner dan pada enzim *hs90a* ketepatan klasifikasi data *training* dengan metode *logistic regression ensembles* (*lorens*) lebih tinggi dibandingkan regresi logirstik biner sedangkan ketepatan klasifikasi pada data *testing* sama antara kedua metode.

Kedua metode memberikan selisih ketepatan klasifikasi yang kecil namun hasil klasifikasi dengan metode *logistic regression ensembles* (*Lorens*) lebih terpercaya karena metode ini menggunakan nilai *threshold* optimal yang tidak terdapat pada regresi logistik biner.

(Halaman ini sengaja dikosongkan)

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut.

1. Berdasarkan hasil deskripsi mengenai jenis enzim pada *database DUD-E* diketahui bahwa enzim *aofb, cah2* dan *hs90a* mempunyai proporsi respon positif (*ligand*) sebesar 25 persen dan respon negatif (*decoy*) sebesar 75 persen. Pada enzim *aofb* 71 persen senyawa atau variabel prediktor terdiri dari tipe struktur spesifik, 21 persen terdiri dari tipe berat dan permukaan, 9 persen terdiri dari tipe *A Log P*,4 persen dari tipe energi dan 6 persen dari tipe lainnya. Pada enzim *cah2* 61 persen senyawa atau variabel prediktor terdiri dari tipe struktur spesifik, 21 persen terdiri dari tipe berat permukaan, 8 persen terdiri dari tipe *A Log P*, 4 persen dari unsur energi dan 6 persen dari tipe lainnya. Pada enzim *hs90a* 59 persen senyawa atau variabel prediktor terdiri dari tipe struktur spesifik, 22 persen terdiri dari tipe berat permukaan, 9 persen terdiri dari tipe *A Log P*, 4 persen dari unsur energi dan 6 persen dari tipe lainnya.
2. Pembagian data *training testing* yang digunakan dalam klasifikasi enzim pada *database DUD-E* dengan regresi logistik biner adalah 90%:10% dan 85%:15%. Dimana kombinasi data *training testing* yang memberikan ketepatan klasifikasi tertinggi adalah kombinasi data training testing 90%:10%. Ketepatan klasifikasi yang dihasilkan pada enzim *aofb* sebesar 90,4 persen untuk data *training* dan 91,04 persen untuk data *testing*. Model optimum terbentuk pada iterasi kedua belas dengan melibatkan 12 variabel prediktor yaitu *ALogP98 Unknown, IsChiral, LogD, HBD Count, Num Explicit Atoms, Num AromaticBonds, Num AromaticRings, Num Chains, Num H Acceptors, Num H Donors, Num H Donors Lipinski* dan

Rad Of Gyration. Ketepatan klasifikasi yang dihasilkan pada enzim *cah2* sebesar 91,7 persen untuk data *training* dan 91,3 persen untuk data *testing*. Model optimum terbentuk pada iterasi kedelapan dengan melibatkan 8 variabel prediktor yaitu *Molecular Mas, HBA Count, N Plus O Count, Num Rings6, Num_Stereo Bonds, Num Atom Classes, Num_H_Acceptors* dan *Molecular 3D Polar SASA* namun terdapat satu variabel yang tidak signifikan terhadap model yaitu variabel *Num_H_Acceptors*. Ketepatan klasifikasi yang dihasilkan pada enzim *hs90a* sebesar 94 persen untuk data *training* dan 100 persen untuk data *testing*. Model optimum terbentuk pada iterasi keempat dengan melibatkan 4 variabel prediktor yaitu *Is Chiral, Num_Rings9Plus, Num H Acceptors dan Rad Of Gyration* namun terdapat dua variabel yang tidak signifikan yaitu variabel *Is Chiral* dan *Num_Rings9Plus*.

3. Analisis klasifikasi menggunakan metode *logistic regression ensembe* dilakukan dengan membagi data menjadi data *training* dan *testing* dengan proporsi 90:10% dan 85%:15%. Kombinasi data *training testing* yang memberikan ketepatan klasifikasi tertinggi adalah kombinasi data *training testing* 90%:10%. Pada enzim *aofb* didapatkan partisi optimal dengan membagi variabel prediktor kedalam 9 sub ruang dan didapatkan ketepatan klasifikasi untuk data *training* sebesar 85,12 dan untuk data *testing* sebesar 89,55 persen. Pada enzim *cah2* didapatkan partisi optimal dengan membagi variabel prediktor kedalam 4 sub ruang dan didapatkan ketepatan klasifikasi untuk data *training* sebesar 94,24 persen dan untuk data *testing* sebesar 92,81 persen. Pada enzim *hs90a* didapatkan partisi optimal dengan membagi variabel prediktor kedalam 5 sub ruang dan didapatkan ketepatan klasifikasi untuk data *training* sebesar 96,22 persen dan untuk data *testing* sebesar 100 persen.

4. Perbandingan kedua metode yaitu regresi logistik biner dan *logistic regression ensemble* dalam klasifikasi enzim pada *database DUD-E* memberikan perbandingan yang berbeda pada setiap enzim. Pada enzim *aofb* ketepatan klasifikasi menggunakan metode regresi logistik biner lebih tinggi dibandingkan dengan ketepatan klasifikasi dengan metode *logistic regression ensemble* baik dengan data *training* maupun data *testing*. Sedangkan pada enzim *cah2* dan *hs90a* ketepatan klasifikasi pada data *training* yang didapatkan dengan metode *logistic regression ensemble* lebih tinggi dibandingkan dengan metode regresi logistik biner dan ketepatan klasifikasi yang sama pada data *testing*. Meskipun kedua metode hanya memberikan selisih ketepatan klasifikasi yang kecil namun hasil klasifikasi dengan metode *logistic regression ensemble* lebih terpercaya karena menggunakan *treshold* optimal yang dapat menyeimbangkan *sensitivity* dan *specitificity* pada data yang tidak mempunyai respon positif dan negatif yang tidak seimbang. Namun metode *Lorens* juga memiliki kelemahan yaitu tidak menghasilkan model yang dapat diinterpretasikan sehingga digunakan regresi logistik biner untuk mendapatkan model yang dapat diinterpretasikan.

5.2 Saran

Saran yang dapat diberikan dari hasil penelitian ini adalah sebagai berikut.

1. Untuk dijadikan sebagai rekomendasi kepada peneliti selanjutnya yaitu penambahan penjelasan mengenai kandungan senyawa yang terkandung pada enzim yang digunakan dalam penelitian.
2. Untuk mengklasifikasikan *high dimensional* data menggunakan metode *logistic regression ensemble* (*Lorens*) sebaiknya digunakan jumlah variabel yang lebih banyak dari pengamatan yang digunakan.
3. Pada penelitian selanjutnya sebaiknya dipelajari metode *Logistic Regression Ensemble* (*Lorens*) menggunakan *fitur*

selection untuk mengetahui keberartian variabel yang digunakan dalam *high dimensional* data.

DAFTAR PUSTAKA

Agresti, A. (1990). *Categorical Data Analysis.* New York: John Wiley and Sons.

Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., & Kodell, R. L. (2006). *Classification by ensemble from random partitions of high-dimensional data. Computational statistic and data analysis*, 4-6.

Champe, P.C.(2007). *Lippincott Illustrated Reviews:Biochemistry 4th edition* . New York: Lippincott Wiliams& Wilkins.

DiMasi, J. A., Hansen, R. W., & Grabowski, H. G. (2002). *The price of innovation: new estimates of drug development costs.*

Fingleton, B. (2007). *Matrix Metalloproteinases as Valid Clinical Targets*. 334.

Firyanto, W. F., Santoso, L. A., & Waspadji, S. (2009). *Peran Heut Shock Protein terhadap Resistensi Insulin*. *Volum: 59*, 122.

Jenwitheesuk, E. H. (2008). *Novel Paradigms for Drug Discovery Computational multitarget screening*. *Trends in Pharmacological Sciences* , 29,62-71.

Lee, K. A. (2013). *Multinomial Logistic Regression Ensembles*. *Biopharm Stat*, 23(3).

Lengauer, T. d. (1996). *Computational Methods for Biomolecular*. *Curr. Opin. Struct. Biol*, 402-406.

Lim, N. (2007). *Classification by Ensembles from Random Partitions using Logistic Models*. In: Applied Matheematics and Statistics. *Stony Brook University*.

Lim, N., Ahn, H., Moon, H., & Chen, J. J. (2010). *Classification High Dimensional Data With Ensemble of Logistic Regression Models*. 2.

Makowski L, H. G. (2004). *Fatty acid binding proteins—the evolutionary crossroads of inflammatory and metabolic responses*. 134:2464S–8S.

Markus, B., & Edgar, J. (2004). *Chemogenomics: an Emerging Strategy*. *Nature Reviews Genetics*, 262-275.

Martono, N. P. (2014). *Customer Lifetime Value And Defection Possibility Prediction Model Using Machine Learning*.

Okada, M., Ohwada, H., & Aoki, S. (2013). *Docking Score Calculation Using Machine Learning With An Enhanced Inhibitor Database*. *Bioinformatics and computational Biology*, 1.

Visse R, N. H. (2003). *Matrix metalloproteinases and tissue inhibitors*. *structure, function and biochemistry*, 92: 827-39.

Watson Hosmer, D., & Lemeshow, S. (1995). *Applied Logistic Regression.* New York: John Wiley.

Widjaja, F. F., Santoso, L. A., & Waspadji, S. (2009). *Peran Heat Shock Protein terhadap Resistensi Insulin.* 121.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *DATA MINING : Practical Machine Learning Tools and Techniques* (3 ed.). Burlington: Morgan Kaufmann.

Yi Lin, T., Kai Liao, B., Lin Horng, J., Jiun Yan, J., Der Hsiao, C., & Pung Hwang, P. (2008). *Carbonic anhydrase 2-like a and 15a are involved in acid-base regulation. 1*.

LAMPIRAN A
Data Pengamatan

Data Pengamatan Enzim *aofb*

No	Y	X_1	X_2	X_3	X_4	X_{70}
1	1	1.473273	0.705604	1.476055	⋯	0.623428
2	1	0.416692	1.140121	0.811798	⋯	-0.4785
3	1	0.416692	1.140121	0.811798	⋯	-0.35888
4	1	0.416692	1.140121	0.811798	⋯	-0.33758
⋮	⋮	⋮	⋮	⋮	⋮	⋮
671	0	1.350869	0.975871	1.353435	0	0.943956
672	0	0.384402	0.725033	0.606428	0	0.741789

Keterangan :

x1	:	ALogP	x36	:	Num_Rings5
x2	:	ALogP_MR	x37	:	Num_Rings6
x3	:	ALogP98	x38	:	Num_Rings7
x4	:	ALogP98_Unknown	x39	:	Num_Rings8
x5	:	Apol	x40	:	Num_Rings9Plus
x6	:	FormalCharge	x41	:	Num_Chains
x7	:	IsChiral	x42	:	Num_ChainAssemblies
x8	:	AverageBondLength	x43	:	Num_StereoAtoms
x9	:	LogD	x44	:	Num_StereoBonds
x10	:	Molecular_Weight	x45	:	Num_AtomClasses
x11	:	Molecular_Mass	x46	:	Num_TerminalRotomers
x12	:	Molecular_Solubility	x47	:	Num_TrueStereoAtoms
x13	:	VSA_TotalArea	x48	:	Num_UnknownTrueStereoAtoms
x14	:	HBA_Count	x49	:	Num_UnknownPseudoStereoAtoms
x15	:	HBD_Count	x50	:	Num_MesoStereoAtoms
x16	:	NPlusO_Count	x51	:	Num_H_Acceptors
x17	:	Num_Atoms	x52	:	Num_H_Donors
x18	:	Num_Bonds	x53	:	Num_H_Acceptors_Lipinski
x19	:	Num_Hydrogens	x54	:	Num_H_Donors_Lipinski
x20	:	Num_ExplicitHydrogens	x55	:	Organic_Count

x21	:	Num_ExplicitAtoms	x56	:	Molecular_Volume
x22	:	Num_ExplicitBonds	x57	:	Molecular_SurfaceArea
x23	:	Num_PositiveAtoms	x58	:	Molecular_PolarSurfaceArea
x24	:	Num_NegativeAtoms	x59	:	Molecular_FractionalPolarSurfaceArea
x25	:	Num_SpiroAtoms	x60	:	Molecular_SASA
x26	:	Num_BridgeHeadAtoms	x61	:	Molecular_PolarSASA
x27	:	Num_RingBonds	x62	:	Molecular_FractionalPolarSASA
x28	:	Num_RotatableBonds	x63	:	Molecular_SAVol
x29	:	Num_AromaticBonds	x64	:	Energy
x30	:	Num_BridgeBonds	x65	:	Minimized_Energy
x31	:	Num_Rings	x66	:	Strain_Energy
x32	:	Num_AromaticRings	x67	:	RadOfGyration
x33	:	Num_RingAssemblies	x68	:	Molecular_3D_SASA
x34	:	Num_Rings3	x69	:	Molecular_3D_PolarSASA
x35	:	Num_Rings4	x70	:	Molecular_3D_SAVol
y	:	klasifikasi enzim			

Data Pengamatan Enzim *cah2*

No	Y	X_1	X_2	X_3	X_4	X_{71}
1	1	1.086641	2.313608	1.010269	⋯	1.804768
2	1	1.086641	2.313608	1.010269	⋯	1.852033
3	1	-0.63817	-0.02432	-0.50802	⋯	0.824015
4	1	1.077311	2.367613	1.001025	⋯	1.787495
⋮	⋮	⋮	⋮	⋮	⋮	⋮
671	0	-0.32097	-0.63436	-0.2723	0	-0.3592
672	0	-0.38919	-0.51194	-0.34394	0	-0.34175

Keterangan

x1	:	ALogP	x37	:	Num_Rings6
x2	:	ALogP_MR	x38	:	Num_Rings7
x3	:	ALogP98	x39	:	Num_Rings8
x4	:	ALogP98_Unknown	x40	:	Num_Rings9Plus
x5	:	Apol	x41	:	Num_Chains
x6	:	FormalCharge	x42	:	Num_ChainAssemblies

x7	:	IsChiral	x43	:	Num_StereoAtoms
x8	:	AverageBondLength	x44	:	Num_StereoBonds
x9	:	LogD	x45	:	Num_UnknownStereoBonds
x10	:	Molecular_Weight	x46	:	Num_AtomClasses
x11	:	Molecular_Mass	x47	:	Num_TerminalRotomers
x12	:	Molecular_Solubility	x48	:	Num_TrueStereoAtoms
x13	:	VSA_TotalArea	x49	:	Num_UnknownTrueStereoAtoms
x14	:	HBA_Count	x50	:	Num_UnknownPseudoStereoAtoms
x15	:	HBD_Count	x51	:	Num_MesoStereoAtoms
x16	:	NPlusO_Count	x52	:	Num_H_Acceptors
x17	:	Num_Atoms	x53	:	Num_H_Donors
x18	:	Num_Bonds	x54	:	Num_H_Acceptors_Lipinski
x19	:	Num_Hydrogens	x55	:	Num_H_Donors_Lipinski
x20	:	Num_ExplicitHydrogens	x56	:	Organic_Count
x21	:	Num_ExplicitAtoms	x57	:	Molecular_Volume
x22	:	Num_ExplicitBonds	x58	:	Molecular_SurfaceArea
x23	:	Num_PositiveAtoms	x59	:	Molecular_PolarSurfaceArea
x24	:	Num_NegativeAtoms	x60	:	Molecular_FractionalPolarSurfaceArea
x25	:	Num_SpiroAtoms	x61	:	Molecular_SASA
x26	:	Num_BridgeHeadAtoms	x62	:	Molecular_PolarSASA
x27	:	Num_RingBonds	x63	:	Molecular_FractionalPolarSASA
x28	:	Num_RotatableBonds	x64	:	Molecular_SAVol
x29	:	Num_AromaticBonds	x65	:	Energy
x30	:	Num_BridgeBonds	x66	:	Minimized_Energy
x31	:	Num_Rings	x67	:	Strain_Energy
x32	:	Num_AromaticRings	x68	:	RadOfGyration
x33	:	Num_RingAssemblies	x69	:	Molecular_3D_SASA
x34	:	Num_Rings3	x70	:	Molecular_3D_PolarSASA

x35	:	Num_Rings4	x71	:	Molecular_3D_SAVol
x36	:	Num_Rings5	y	:	klasifikasi enzim

Data Pengamatan Enzim *hs90a*

No	Y	X_1	X_2	X_3	X_4	X_{69}
1	1	1.369641	-0.46542	1.262291	⋯	-0.24019
2	1	1.319808	-0.63646	1.21226	⋯	-0.36924
3	1	1.049286	-0.73023	1.137808	⋯	-0.84484
4	1	0.739016	-1.00482	0.825707	⋯	-1.39483
⋮	⋮	⋮	⋮	⋮	⋮	⋮
499	0	0.891481	-0.83591	0.96508	⋯	-0.68883
500	0	0.403236	-1.62277	0.375424	⋯	-2.06239

Keterangan

x1	:	ALogP	x36	:	Num_Rings5
x2	:	ALogP_MR	x37	:	Num_Rings6
x3	:	ALogP98	x38	:	Num_Rings7
x4	:	ALogP98_Unknown	x39	:	Num_Rings9Plus
x5	:	Apol	x40	:	Num_Chains
x6	:	FormalCharge	x41	:	Num_ChainAssemblies
x7	:	IsChiral	x42	:	Num_StereoAtoms
x8	:	AverageBondLength	x43	:	Num_StereoBonds
x9	:	LogD	x44	:	Num_AtomClasses
x10	:	Molecular_Weight	x45	:	Num_TerminalRotomers
x11	:	Molecular_Mass	x46	:	Num_TrueStereoAtoms
x12	:	Molecular_Solubility	x47	:	Num_UnknownTrueStereoAtoms
x13	:	VSA_TotalArea	x48	:	Num_UnknownPseudoStereoAtoms
x14	:	HBA_Count	x49	:	Num_MesoStereoAtoms
x15	:	HBD_Count	x50	:	Num_H_Acceptors
x16	:	NPlusO_Count	x51	:	Num_H_Donors
x17	:	Num_Atoms	x52	:	Num_H_Acceptors_Lipinski
x18	:	Num_Bonds	x53	:	Num_H_Donors_Lipinski
x19	:	Num_Hydrogens	x54	:	Organic_Count

x20	:	Num_ExplicitHydrogens	x55	:	Molecular_Volume
x21	:	Num_ExplicitAtoms	x56	:	Molecular_SurfaceArea
x22	:	Num_ExplicitBonds	x57	:	Molecular_PolarSurfaceArea
x23	:	Num_PositiveAtoms	x58	:	Molecular_FractionalPolarSurfaceArea
x24	:	Num_NegativeAtoms	x59	:	Molecular_SASA
x25	:	Num_SpiroAtoms	x60	:	Molecular_PolarSASA
x26	:	Num_BridgeHeadAtoms	x61	:	Molecular_FractionalPolarSASA
x27	:	Num_RingBonds	x62	:	Molecular_SAVol
x28	:	Num_RotatableBonds	x63	:	Energy
x29	:	Num_AromaticBonds	x64	:	Minimized_Energy
x30	:	Num_BridgeBonds	x65	:	Strain_Energy
x31	:	Num_Rings	x66	:	RadOfGyration
x32	:	Num_AromaticRings	x67	:	Molecular_3D_SASA
x33	:	Num_RingAssemblies	x68	:	Molecular_3D_PolarSASA
x34	:	Num_Rings3	x69	:	Molecular_3D_SAVol
x35	:	Num_Rings4	y	:	klasifikasi enzim

LAMPIRAN B

Output Regresi Logistik Biner dengan Kombinasi Data *Training* 90% dan Data *Testing*10%

Output Regresi Logistik Biner untuk Enzim *aofb*

Case Processing Summary

Unweighted Cases[a]		N	Percent
Selected Cases	Included in Analysis	605	100.0
	Missing Cases	0	.0
	Total	605	100.0
Unselected Cases		0	.0
Total		605	100.0

a. If weight is in effect, see classification table for the total number of cases.

Iteration History[a,b,c]

Iteration		-2 Log likelihood	Coefficients Constant
Step 0	1	674.469	-1.021
	2	673.192	-1.125
	3	673.191	-1.127
	4	673.191	-1.127

a. Constant is included in the model.

b. Initial -2 Log Likelihood: 673.191

c. Estimation terminated at iteration number 4 because parameter estimates changed by less than .001.

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	532.576[a]	.207	.309

2	478.721[a]	.275	.409
3	410.714[b]	.352	.524
4	366.803[b]	.397	.592
5	352.826[b]	.411	.612
6	341.881[b]	.422	.628
7	329.905[b]	.433	.645
8	315.864[c]	.446	.664
9	305.003[c]	.456	.679
10	297.405[c]	.463	.689
11	291.015[c]	.468	.698
12	285.543[c]	.473	.705

a. Estimation terminated at iteration number 5 because parameter estimates changed by less than .001.

b. Estimation terminated at iteration number 6 because parameter estimates changed by less than .001.

c. Estimation terminated at iteration number 7 because parameter estimates changed by less than .001.

Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	281.718	3	.000
2	24.675	8	.002
3	9.322	8	.316
4	9.419	8	.308
5	9.340	8	.314
6	13.001	8	.112
7	13.265	8	.103
8	19.626	8	.012
9	21.737	8	.005
10	21.389	8	.006
11	26.087	8	.001
12	28.789	8	.000

Classification Table[a]

Observed			Predicted		
			y		Percentage Correct
			0	1	
Step 1	y	0	411	46	89.9
		1	62	86	58.1
	Overall Percentage				82.1
Step 2	y	0	441	16	96.5
		1	81	67	45.3
	Overall Percentage				84.0
Step 3	y	0	427	30	93.4
		1	62	86	58.1
	Overall Percentage				84.8
Step 4	y	0	427	30	93.4
		1	54	94	63.5
	Overall Percentage				86.1
Step 5	y	0	429	28	93.9
		1	45	103	69.6
	Overall Percentage				87.9
Step 6	y	0	434	23	95.0
		1	41	107	72.3
	Overall Percentage				89.4
Step 7	y	0	438	19	95.8
		1	39	109	73.6
	Overall Percentage				90.4
Step 8	y	0	436	21	95.4
		1	42	106	71.6
	Overall Percentage				89.6
Step 9	y	0	439	18	96.1
		1	39	109	73.6

	Overall Percentage				90.6
Step 10	y	0	436	21	95.4
		1	41	107	72.3
	Overall Percentage				89.8
Step 11	y	0	436	21	95.4
		1	38	110	74.3
	Overall Percentage				90.2
Step 12	y	0	436	21	95.4
		1	37	111	75.0
	Overall Percentage				90.4

a. The cut value is .500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)	95.0% C.I.for EXP(B)	
								Lower	Upper
Step 1[a]	x54	-1.240	.122	103.869	1	.000	.289	.228	.367
	Constant	-1.411	.118	144.051	1	.000	.244		
Step 2[b]	x41	-.862	.128	45.223	1	.000	.422	.329	.543
	x54	-1.372	.140	96.387	1	.000	.254	.193	.334
	Constant	-1.610	.134	143.820	1	.000	.200		
Step 3[c]	x21	2.964	.403	54.009	1	.000	19.370	8.787	42.696
	x41	-3.760	.437	74.100	1	.000	.023	.010	.055
	x54	-.952	.148	41.145	1	.000	.386	.289	.516
	Constant	-1.867	.161	135.073	1	.000	.155		
Step 4[d]	x8	-.996	.170	34.201	1	.000	.369	.265	.516
	x21	4.295	.486	78.026	1	.000	73.363	28.285	190.283
	x41	-5.423	.550	97.252	1	.000	.004	.002	.013
	x54	-1.050	.159	43.574	1	.000	.350	.256	.478
	Constant	-1.987	.173	132.305	1	.000	.137		
Step 5[e]	x7	.575	.162	12.603	1	.000	1.778	1.294	2.442
	x8	-1.118	.180	38.426	1	.000	.327	.230	.465
	x21	4.815	.531	82.133	1	.000	123.322	43.533	349.352
	x41	-6.277	.635	97.759	1	.000	.002	.001	.007
	x54	-1.214	.169	51.482	1	.000	.297	.213	.414

	Constant	-2.118	.188	126.786	1	.000	.120		
Step 6[f]	x4	.418	.129	10.517	1	.001	1.519	1.180	1.955
	x7	.573	.166	11.934	1	.001	1.774	1.281	2.456
	x8	-1.081	.183	34.967	1	.000	.339	.237	.485
	x21	4.816	.538	80.150	1	.000	123.462	43.017	354.342
	x41	-6.228	.644	93.630	1	.000	.002	.001	.007
	x54	-1.260	.173	53.086	1	.000	.284	.202	.398
	Constant	-2.145	.192	124.739	1	.000	.117		
Step 7[g]	x4	.445	.131	11.462	1	.001	1.560	1.206	2.018
	x7	.723	.178	16.466	1	.000	2.060	1.453	2.921
	x8	-1.185	.191	38.361	1	.000	.306	.210	.445
	x21	3.316	.683	23.589	1	.000	27.552	7.228	105.032
	x41	-5.431	.677	64.451	1	.000	.004	.001	.016
	x54	-1.219	.175	48.499	1	.000	.296	.210	.417
	x67	.994	.298	11.101	1	.001	2.702	1.506	4.849
	Constant	-2.186	.196	123.963	1	.000	.112		
Step 8[h]	x4	.495	.145	11.613	1	.001	1.640	1.234	2.181
	x7	.713	.182	15.418	1	.000	2.040	1.429	2.912
	x8	-1.095	.197	30.755	1	.000	.335	.227	.493
	x21	3.271	.698	21.970	1	.000	26.345	6.709	103.459
	x41	-5.676	.713	63.409	1	.000	.003	.001	.014
	x51	-.810	.226	12.855	1	.000	.445	.286	.693
	x54	-1.492	.200	55.531	1	.000	.225	.152	.333

	x67	1.551	.343	20.396	1	.000	4.717	2.406	9.246
	Constant	-2.275	.207	121.326	1	.000	.103		
Step 9[i]	x4	.622	.153	16.577	1	.000	1.863	1.381	2.513
	x7	.691	.184	14.174	1	.000	1.996	1.393	2.861
	x8	-1.239	.211	34.591	1	.000	.290	.192	.438
	x21	4.046	.773	27.374	1	.000	57.151	12.555	260.151
	x41	-6.354	.784	65.720	1	.000	.002	.000	.008
	x51	-.763	.227	11.240	1	.001	.466	.299	.729
	x52	-.740	.233	10.071	1	.002	.477	.302	.753
	x54	-.982	.247	15.798	1	.000	.375	.231	.608
	x67	1.576	.347	20.607	1	.000	4.837	2.449	9.554
	Constant	-2.303	.214	115.657	1	.000	.100		
Step 10[j]	x4	.560	.150	13.927	1	.000	1.750	1.304	2.348
	x7	.632	.187	11.380	1	.001	1.882	1.303	2.717
	x8	-1.263	.217	33.770	1	.000	.283	.185	.433
	x21	5.560	1.002	30.760	1	.000	259.784	36.417	1.853E3
	x32	-.698	.258	7.322	1	.007	.497	.300	.825
	x41	-7.970	1.048	57.885	1	.000	.000	.000	.003
	x51	-.903	.238	14.467	1	.000	.405	.254	.645
	x52	-.793	.237	11.208	1	.001	.453	.285	.720
	x54	-.917	.253	13.188	1	.000	.400	.244	.656
	x67	1.694	.353	22.966	1	.000	5.439	2.721	10.872
	Constant	-2.329	.219	112.773	1	.000	.097		

Step 11[k]	x4	.548	.152	12.938	1	.000	1.729	1.283	2.330
	x7	.578	.190	9.307	1	.002	1.783	1.230	2.585
	x8	-1.223	.214	32.592	1	.000	.294	.193	.448
	x15	.818	.330	6.160	1	.013	2.267	1.188	4.326
	x21	6.191	1.077	33.069	1	.000	488.338	59.201	4.028E3
	x32	-.788	.267	8.701	1	.003	.455	.269	.768
	x41	-8.276	1.085	58.135	1	.000	.000	.000	.002
	x51	-1.263	.286	19.490	1	.000	.283	.162	.496
	x52	-1.660	.434	14.637	1	.000	.190	.081	.445
	x54	-.650	.279	5.418	1	.020	.522	.302	.902
	x67	1.630	.361	20.415	1	.000	5.105	2.517	10.353
	Constant	-2.420	.230	110.301	1	.000	.089		
Step 12[l]	x4	.586	.156	14.134	1	.000	1.796	1.323	2.437
	x7	.513	.190	7.329	1	.007	1.671	1.152	2.422
	x8	-1.041	.222	21.925	1	.000	.353	.228	.546
	x15	.887	.336	6.966	1	.008	2.428	1.256	4.690
	x21	5.584	1.111	25.269	1	.000	266.038	30.161	2.347E3
	x29	3.604	1.561	5.333	1	.021	36.745	1.725	782.751
	x32	-4.179	1.503	7.736	1	.005	.015	.001	.291
	x41	-7.727	1.117	47.876	1	.000	.000	.000	.004
	x51	-1.454	.304	22.905	1	.000	.234	.129	.424
	x52	-1.535	.439	12.214	1	.000	.215	.091	.510

	x54	-.817	.295	7.680	1	.006	.442	.248	.787
	x67	1.649	.361	20.894	1	.000	5.204	2.566	10.556
	Constant	-2.462	.236	109.023	1	.000	.085		

a. Variable(s) entered on step 1: x54.

b. Variable(s) entered on step 2: x41.

c. Variable(s) entered on step 3: x21.

d. Variable(s) entered on step 4: x8.

e. Variable(s) entered on step 5: x7.

f. Variable(s) entered on step 6: x4.

g. Variable(s) entered on step 7: x67.

h. Variable(s) entered on step 8: x51.

i. Variable(s) entered on step 9: x52.

j. Variable(s) entered on step 10: x32.

k. Variable(s) entered on step 11: x15.

l. Variable(s) entered on step 12: x29.

Output Regresi Logistik Biner untuk Enzim *cah2*

Case Processing Summary

Unweighted Cases[a]		N	Percent
Selected Cases	Included in Analysis	3006	100.0
	Missing Cases	0	.0
	Total	3006	100.0
Unselected Cases		0	.0
Total		3006	100.0

a. If weight is in effect, see classification table for the total number of cases.

Iteration History[a,b,c]

Iteration		-2 Log likelihood	Coefficients Constant
Step 0	1	3363.560	-1.014
	2	3357.495	-1.115
	3	3357.492	-1.117
	4	3357.492	-1.117

a. Constant is included in the model.

b. Initial -2 Log Likelihood: 3357.492

c. Estimation terminated at iteration number 4 because parameter estimates changed by less than .001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	657.311	1	.000
	Block	657.311	1	.000
	Model	657.311	1	.000
Step 2	Step	309.299	1	.000
	Block	966.610	2	.000

	Model	966.610	2	.000
Step 3	Step	305.636	1	.000
	Block	1272.246	3	.000
	Model	1272.246	3	.000
Step 4	Step	143.908	1	.000
	Block	1416.154	4	.000
	Model	1416.154	4	.000
Step 5	Step	191.223	1	.000
	Block	1607.377	5	.000
	Model	1607.377	5	.000
Step 6	Step	203.060	1	.000
	Block	1810.437	6	.000
	Model	1810.437	6	.000
Step 7	Step	98.560	1	.000
	Block	1908.998	7	.000
	Model	1908.998	7	.000
Step 8	Step	155.958	1	.000
	Block	2064.955	8	.000
	Model	2064.955	8	.000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	2700.181[a]	.196	.292
2	2390.882[a]	.275	.409
3	2085.246[b]	.345	.513
4	1941.338[b]	.376	.558
5	1750.115[b]	.414	.616
6	1547.055[c]	.452	.673
7	1448.494[c]	.470	.699
8	1292.536[d]	.497	.739

a. Estimation terminated at iteration number 5 because parameter estimates changed by less than .001.

b. Estimation terminated at iteration number 6 because parameter estimates changed by less than .001.

c. Estimation terminated at iteration number 7 because parameter estimates changed by less than .001.

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	36.972	8	.000
2	44.173	8	.000
3	18.628	8	.017
4	67.760	8	.000
5	38.108	8	.000
6	22.550	8	.004
7	24.902	8	.002
8	45.022	8	.000

Classification Table[a]

			Predicted		
			y		
	Observed		0	1	Percentage Correct
Step 1	y	0	2137	128	94.3
		1	464	277	37.4
	Overall Percentage				80.3
Step 2	y	0	2125	140	93.8
		1	385	356	48.0
	Overall Percentage				82.5

Step 3	y	0	2139	126	94.4
		1	333	408	55.1
	Overall Percentage				84.7
Step 4	y	0	2130	135	94.0
		1	285	456	61.5
	Overall Percentage				86.0
Step 5	y	0	2133	132	94.2
		1	236	505	68.2
	Overall Percentage				87.8
Step 6	y	0	2149	116	94.9
		1	206	535	72.2
	Overall Percentage				89.3
Step 7	y	0	2158	107	95.3
		1	188	553	74.6
	Overall Percentage				90.2
Step 8	y	0	2169	96	95.8
		1	154	587	79.2
	Overall Percentage				91.7

a. The cut value is .500

Variables in the Equation[i]

		B	S.E.	Wald	df	Sig.	Exp(B)	95.0% C.I.for EXP(B)	
								Lower	Upper
Step 1[a]	x70	1.221	.056	477.577	1	.000	3.389	3.038	3.781
	Constant	-1.373	.052	704.913	1	.000	.253		
Step 2[b]	x16	-1.333	.086	243.007	1	.000	.264	.223	.312
	x70	2.292	.097	560.451	1	.000	9.893	8.183	11.960
	Constant	-1.569	.059	696.589	1	.000	.208		
Step 3[c]	x14	1.367	.087	247.825	1	.000	3.923	3.309	4.651
	x16	-2.413	.121	399.855	1	.000	.090	.071	.113
	x70	2.497	.108	536.946	1	.000	12.152	9.838	15.010
	Constant	-1.762	.067	686.853	1	.000	.172		
Step 4[d]	x14	1.443	.092	244.880	1	.000	4.233	3.533	5.071
	x16	-2.651	.127	432.973	1	.000	.071	.055	.091
	x37	.711	.063	128.797	1	.000	2.036	1.800	2.302
	x70	2.728	.116	556.141	1	.000	15.296	12.193	19.187
	Constant	-1.893	.073	667.897	1	.000	.151		
Step 5[e]	x14	1.466	.095	238.471	1	.000	4.331	3.596	5.216
	x16	-2.009	.132	232.348	1	.000	.134	.104	.174
	x37	1.217	.080	233.794	1	.000	3.377	2.889	3.947
	x46	-1.194	.093	165.610	1	.000	.303	.253	.363
	x70	2.749	.123	500.892	1	.000	15.634	12.289	19.890

	Constant	-2.091	.083	640.263	1	.000	.124		
Step 6[f]	x14	1.592	.103	240.241	1	.000	4.915	4.018	6.011
	x16	-2.093	.143	213.388	1	.000	.123	.093	.163
	x37	1.445	.086	281.360	1	.000	4.244	3.584	5.025
	x44	-1.199	.100	142.845	1	.000	.302	.248	.367
	x46	-1.261	.100	160.325	1	.000	.283	.233	.344
	x70	2.948	.134	482.411	1	.000	19.067	14.657	24.804
	Constant	-2.326	.094	613.871	1	.000	.098		
Step 7[g]	x11	1.393	.147	89.780	1	.000	4.028	3.020	5.374
	x14	1.499	.108	191.151	1	.000	4.478	3.620	5.538
	x16	-2.231	.155	206.870	1	.000	.107	.079	.146
	x37	1.162	.091	161.859	1	.000	3.196	2.672	3.822
	x44	-1.129	.106	112.707	1	.000	.323	.262	.398
	x46	-2.099	.141	220.616	1	.000	.123	.093	.162
	x70	2.708	.136	396.484	1	.000	15.003	11.492	19.586
	Constant	-2.312	.095	588.831	1	.000	.099		
Step 8[h]	x11	2.346	.181	168.725	1	.000	10.445	7.331	14.882
	x14	1.369	.113	147.752	1	.000	3.930	3.152	4.901
	x16	-2.032	.161	158.830	1	.000	.131	.096	.180
	x37	1.316	.098	179.631	1	.000	3.729	3.076	4.520
	x44	-1.116	.111	101.976	1	.000	.328	.264	.407
	x46	-3.194	.186	294.515	1	.000	.041	.028	.059

x51	-2.913	524.555	.000	1	.996	.054	.000	.
x70	2.684	.143	352.392	1	.000	14.639	11.062	19.373
Constant	-2.634	47.210	.003	1	.956	.072		

a. Variable(s) entered on step 1: x70.

b. Variable(s) entered on step 2: x16.

c. Variable(s) entered on step 3: x14.

d. Variable(s) entered on step 4: x37.

e. Variable(s) entered on step 5: x46.

f. Variable(s) entered on step 6: x44.

g. Variable(s) entered on step 7: x11.

h. Variable(s) entered on step 8: x51.

i. Stepwise procedure stopped because removing the least significant variable result in a previously fitted model.

Output Regresi Logistik Biner untuk Enzim *hs90a*

Case Processing Summary

Unweighted Cases[a]		N	Percent
Selected Cases	Included in Analysis	450	100.0
	Missing Cases	0	.0
	Total	450	100.0
Unselected Cases		0	.0
Total		450	100.0

a. If weight is in effect, see classification table for the total number of cases.

Iteration History[a,b,c]

Step 0	1	508.009	-.996
	2	507.198	-1.090
	3	507.197	-1.093
	4	507.197	-1.093

a. Constant is included in the model.

b. Initial -2 Log Likelihood: 507.197

c. Estimation terminated at iteration number 4 because parameter estimates changed by less than .001.

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	364.974[a]	.271	.401
2	262.736[b]	.419	.620
3	179.192[c]	.518	.766
4	136.524[d]	.561	.830

a. Estimation terminated at iteration number 5 because parameter estimates changed by less than .001.

b. Estimation terminated at iteration number 6 because parameter estimates changed by less than .001.

c. Estimation terminated at iteration number 8 because parameter estimates changed by less than .001.

d. Estimation terminated at iteration number 20 because maximum iterations has been reached. Final solution cannot be found.

Hosmer and Lemeshow Test

Step	Chi-square	df	Sig.
1	26.047	4	.000
2	23.609	7	.001
3	6.386	8	.604
4	1.748	8	.988

Classification Table[a]

Observed			Predicted		
			y		Percentage Correct
			0	1	
Step 1	y	0	312	25	92.6
		1	72	41	36.3
	Overall Percentage				78.4
Step 2	y	0	300	37	89.0
		1	23	90	79.6
	Overall Percentage				86.7
Step 3	y	0	315	22	93.5
		1	14	99	87.6
	Overall Percentage				92.0
Step 4	y	0	319	18	94.7
		1	9	104	92.0
	Overall Percentage				94.0

a. The cut value is .500

Variables in the Equation[e]

		B	S.E.	Wald	df	Sig.	Exp(B)	95.0% C.I.for EXP(B)	
								Lower	Upper
Step 1[a]	x50	1.580	.167	89.117	1	.000	4.854	3.496	6.738
	Constant	-1.440	.144	100.016	1	.000	.237		
Step 2[b]	x7	-1.601	.202	62.852	1	.000	.202	.136	.300
	x50	1.696	.201	71.251	1	.000	5.451	3.677	8.082
	Constant	-2.312	.260	78.823	1	.000	.099		
Step 3[c]	x7	-1.876	.259	52.477	1	.000	.153	.092	.255
	x50	2.219	.283	61.420	1	.000	9.199	5.281	16.023
	x66	-1.921	.288	44.499	1	.000	.146	.083	.257
	Constant	-3.601	.453	63.123	1	.000	.027		
Step 4[d]	x7	-10.615	1.005E3	.000	1	.992	.000	.000	.
	x39	7.968	1.960E3	.000	1	.997	2.886E3	.000	.
	x50	1.903	.345	30.430	1	.000	6.706	3.410	13.185
	x66	-1.981	.330	36.004	1	.000	.138	.072	.263
	Constant	-12.469	1.209E3	.000	1	.992	.000		

a. Variable(s) entered on step 1: x50.

b. Variable(s) entered on step 2: x7.

c. Variable(s) entered on step 3: x66.

d. Variable(s) entered on step 4: x39.

e. Stepwise procedure stopped because removing the least significant variable result in a previously fitted model.

LAMPIRAN C

***Output* Regresi Logistik Biner dengan Kombinasi Data *Training* 85% dan Data *Testing*15%**

Ketepatan kasifikasi untuk enzim *aofb*

Classification Table[a]

			Predicted		
			VAR00071		Percentage Correct
	Observed		0	1	
Step 1	VAR00071	0	385	45	89.5
		1	57	84	59.6
	Overall Percentage				82.1
Step 2	VAR00071	0	415	15	96.5
		1	75	66	46.8
	Overall Percentage				84.2
Step 3	VAR00071	0	403	27	93.7
		1	57	84	59.6
	Overall Percentage				85.3
Step 4	VAR00071	0	401	29	93.3
		1	51	90	63.8
	Overall Percentage				86.0
Step 5	VAR00071	0	404	26	94.0
		1	42	99	70.2
	Overall Percentage				88.1
Step 6	VAR00071	0	410	20	95.3
		1	40	101	71.6
	Overall Percentage				89.5
Step 7	VAR00071	0	411	19	95.6
		1	39	102	72.3
	Overall Percentage				89.8
Step 8	VAR00071	0	408	22	94.9

		1	41	100	70.9
	Overall Percentage				89.0
Step 9	VAR00071	0	411	19	95.6
		1	38	103	73.0
	Overall Percentage				90.0
Step 10	VAR00071	0	411	19	95.6
		1	37	104	73.8
	Overall Percentage				90.2
Step 11	VAR00071	0	412	18	95.8
		1	38	103	73.0
	Overall Percentage				90.2
Step 12	VAR00071	0	412	18	95.8
		1	33	108	76.6
	Overall Percentage				91.1

a. The cut value is .500

Ketepatan kasifikasi untuk enzim *cah2*

Classification Table[a]

			Predicted		
			VAR00072		
	Observed		0	1	Percentage Correct
Step 1	VAR00072	0	2012	123	94.2
		1	439	265	37.6
	Overall Percentage				80.2
Step 2	VAR00072	0	1997	138	93.5
		1	363	341	48.4
	Overall Percentage				82.4
Step 3	VAR00072	0	2011	124	94.2
		1	317	387	55.0

	Overall Percentage				84.5
Step 4	VAR00072	0	2002	133	93.8
		1	276	428	60.8
	Overall Percentage				85.6
Step 5	VAR00072	0	2009	126	94.1
		1	227	477	67.8
	Overall Percentage				87.6
Step 6	VAR00072	0	2026	109	94.9
		1	193	511	72.6
	Overall Percentage				89.4
Step 7	VAR00072	0	2028	107	95.0
		1	161	543	77.1
	Overall Percentage				90.6
Step 8	VAR00072	0	2033	102	95.2
		1	157	547	77.7
	Overall Percentage				90.9
Step 9	VAR00072	0	2044	91	95.7
		1	149	555	78.8
	Overall Percentage				91.5
Step 10	VAR00072	0	2052	83	96.1
		1	132	572	81.2
	Overall Percentage				92.4

a. The cut value is .500

Ketepatan kasifikasi untuk enzim *hs90a*

Classification Table[a]

Observed			Predicted		
			VAR00070		Percentage Correct
			0	1	
Step 1	VAR00070	0	295	23	92.8
		1	68	39	36.4
	Overall Percentage				78.6
Step 2	VAR00070	0	283	35	89.0
		1	20	87	81.3
	Overall Percentage				87.1
Step 3	VAR00070	0	299	19	94.0
		1	11	96	89.7
	Overall Percentage				92.9
Step 4	VAR00070	0	300	18	94.3
		1	9	98	91.6
	Overall Percentage				93.6

a. The cut value is .500

LAMPIRAN D
Output R dari *Logistic Regression Ensemble* (*Lorens*)

Output R dari *Logistic Regression Ensemble* pada Enzim *aofb*(Partisi 9)
1. Output Random Partisi

Variabel	ens1	ens2	ens3	ens4	ens5	ens6	ens7	ens8	ens9	ens10
ALogP	1	1	8	9	5	6	9	3	1	9
ALogP_MR	5	9	8	7	9	8	9	3	9	4
ALogP98	9	3	9	1	8	1	8	1	7	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	8	2	7	8	7	9	4	2	7	6
Molecular_3D_SAVol	7	9	9	6	3	1	8	1	6	2

2. Output koefisien Variabel

Variabel	ens1	ens2	ens3	ens4	ens5
ALogP	0.501	0.676	0.302	0.400	0.930
ALogP_MR	1.852	0.166	2.078	0.597	-1.401
ALogP98	0.809	1.157	0.619	1.512	1.173
⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	-0.175	-0.272	-0.634	0.021	-0.788
Molecular_3D_SAVol	1.783714	0.799	-0.972	1.828	-0.478

3. Output koefisien Variabel(Lanjutan)

Variabel	ens6	ens7	ens8	ens9	ens10
ALogP	0.839	0.948	0.693	0.841	0.448
ALogP_MR	2.787	-0.437	1.666	2.047	-0.892
ALogP98	1.028	0.969	0.962	1.187	1.041
⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	-0.342	0.051	0.088	-0.326	-0.492
Molecular_3D_SAVol	1.848	1.658	2.279	1.912	3.008

4. Output Intersep

Partisi	ens1	ens2	ens3	ens4	ens5	ens6	ens7	ens8	ens9	ens10
[1,]	-1.83531	-1.95551	-1.57764	-1.76868	-2.41558	-1.85603	-2.21791	-1.74239	-2.14269	-1.94338
[2,]	-1.61137	-1.42356	-1.71779	-2.20812	-1.84299	-2.22823	-2.53211	-1.93436	-1.79601	-1.8897
[3,]	-1.46181	-2.48896	-1.75522	-1.92544	-1.56157	-1.39649	-1.86531	-2.04862	-2.04197	-1.92352
[4,]	-2.34534	-1.72251	-1.81491	-2.17252	-2.21076	-1.73342	-1.52451	-1.56208	-1.83806	-1.41452
[5,]	-1.79381	-1.73513	-1.99059	-1.49848	-2.07598	-2.03463	-1.76098	-1.87648	-2.09582	-1.65759
[6,]	-1.78055	-1.51328	-1.9157	-1.5701	-1.66224	-2.5646	-1.42426	-1.62226	-1.66951	-1.87283
[7,]	-1.42644	-1.81691	-1.94302	-1.28137	-1.8165	-1.49739	-1.58409	-1.81704	-1.60823	-1.66291
[8,]	-1.63269	-1.98809	-1.90126	-1.57758	-1.6739	-1.78829	-1.5167	-1.83685	-1.5642	-1.50957
[9,]	-2.00722	-1.78589	-1.62917	-1.9388	-1.41037	-1.52432	-1.74144	-1.99808	-1.75068	-1.63204

5. Output Ketepatan Klasifikasi Data *Training*

	pred.pos	pred.neg
real.pos	99	49
real.neg	41	416

6. Output Ketepatan Klasifikasi Data *Testing*

	pred.pos	pred.neg
real.pos	15	5
real.neg	2	45

Output R dari *Logistic Regression Ensemble* pada Enzim *cah2* (Partisi 4)

1. Output Random Partisi

Variabel	ens1	ens2	ens3	ens4	ens5	ens6	ens7	ens8	ens9	ens10
ALogP	1	3	1	2	1	2	2	3	1	4
ALogP_MR	3	3	4	1	3	2	1	4	4	2
ALogP98	1	2	1	1	3	3	2	1	2	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	2	1	2	4	4	2	1	1	4	4
Molecular_3D_SAVol	4	2	3	3	3	3	4	3	1	2

2. Output koefisien Variabel

Variabel	ens1	ens2	ens3	ens4	ens5
ALogP	1.071	-0.360	0.291	0.511	0.153
ALogP_MR	-2.615	0.354	0.531	-1.787	-8.502
ALogP98	-0.355	0.286	0.889	0.185	1.547
⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	3.091	1.928	1.115	2.270	2.738
Molecular_3D_SAVol	1.746	-6.723	-4.709	0.145	3.928

3. Output koefisien Variabel(Lanjutan)

Variabel	ens6	ens7	ens8	ens9	ens10
ALogP	2.150	0.838	-0.286	0.323	0.247
ALogP_MR	-2.641	-0.109	10.076	-1.475	0.493
ALogP98	0.199	-1.599	0.882	0.501	0.271
⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	1.778	2.020	2.176	1.610	2.426
Molecular_3D_SAVol	1.228	-9.805	1.394	-9.817	-14.351

4. Output Intersep

	ens1	ens2	ens3	ens4	ens5	ens6	ens7	ens8	ens9	ens10
[1,]	-2.959	-2.201	-2.214	-1.858	-1.938	-2.412	-2.307	-3.027	-19.039	-1.988
[2,]	-2.262	-2.067	-3.463	-2.413	-3.132	-2.068	-2.466	-2.413	-2.419	-1.573

[3,]	-1.668	-3.125	-2.299	-2.029	-2.904	-1.892	-1.822	-1.758	-2.232	-2.104
[4,]	-2.115	-1.841	-1.947	-2.188	-1.972	-2.518	-2.021	-2.669	-2.589	-2.542

5. Output Ketepatan Klasifikasi Data *Training*

	pred.pos	pred.neg
real.pos	668	73
real.neg	100	2165

6. Output Ketepatan Klasifikasi Data *Testing*

	pred.pos	pred.neg
real.pos	79	15
real.neg	9	231

Output R dari *Logistic Regression Ensemble* pada Enzim *hs90a* (Partisi 5)

1. Output Random Partisi

Variabel	ens1	ens2	ens3	ens4	ens5	ens6	ens7	ens8	ens9	ens10
ALogP	5	4	3	4	5	2	2	2	2	5
ALogP_MR	2	3	4	3	5	4	3	3	3	4
ALogP98	4	3	1	2	4	5	2	2	3	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	3	1	5	3	3	5	3	2	4	2
Molecular_3D_SAVol	4	1	5	5	3	2	1	3	2	2

2. Output koefisien Variabel

Variabel	ens1	ens2	ens3	ens4	ens5
ALogP	1.690	1.254	0.302	1.466	0.995
ALogP_MR	-5.146	0.004	-8.206	-1.996	-3.831
ALogP98	0.672	0.081	-0.434	0.827	1.429
⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	1.638	2.000	3.105	1.576	2.566
Molecular_3D_SAVol	-1.578	-3.565	-2.710	2.968	-7.865

3. Output koefisien Variabel(Lanjutan)

Variabel	ens6	ens7	ens8	ens9	ens10
ALogP	1.115	2.090	2.055	1.878	-0.663
ALogP_MR	-0.593	-0.512	-2.848	-2.218	-0.008
ALogP98	0.337	-1.937	-0.534	1.027	-1.967
⋮	⋮	⋮	⋮	⋮	⋮
Molecular_3D_PolarSASA	1.205	2.784	1.812	2.031	0.690
Molecular_3D_SAVol	0.235	-2.169	-1.424	-1.312	-0.143

4. Tabel Intersep

	ens1	ens2	ens3	ens4	ens5	ens6	ens7	ens8	ens9	ens10
[1,]	-4.657	-4.974	-5.254	-2.297	-2.625	-6.859	-2.481	-3.338	-4.039	-5.161
[2,]	-4.662	-4.572	-22.212	-3.228	-2.904	-4.019	-11.896	-4.163	-3.304	-4.531

[3,]	-4.602	-1.581	-3.325	-4.191	-4.793	-4.897	-7.265	-1.875	-4.184	-3.472
[4,]	-2.887	-2.804	-7.693	-4.641	-5.040	-2.361	-6.638	-3.305	-4.723	-2.465
[5,]	-2.334	-2.971	-3.013	-13.502	-4.032	-0.673	-3.600	-25.137	-4.434	-20.369

5. Output Ketepatan Klasifikasi Data *Training*

	pred.pos	pred.neg
real.pos	111	2
real.neg	15	322

6. Output Ketepatan Klasifikasi Data *Testing*

	pred.pos	pred.neg
real.pos	12	0
real.neg	0	38

Output R dari *Logistic Regression Ensemble* dengan proporsi data training testing 85%:10%

Ketepatan klasifikasi untuk enzim *aofb*

1. Output Ketepatan Klasifikasi Data *Training*

	pred.pos	pred.neg
real.pos	96	45
real.neg	35	395

2. Output Ketepatan Klasifikasi Data *Testing*

	pred.pos	pred.neg
real.pos	17	10
real.neg	5	69

Ketepatan klasifikasi untuk enzim *cah2*

1. Output Ketepatan Klasifikasi Data *Training*

	pred.pos	pred.neg
real.pos	639	65
real.neg	89	2046

2. Output Ketepatan Klasifikasi Data *Testing*

	pred.pos	pred.neg
real.pos	113	18
real.neg	14	356

Ketepatan klasifikasi untuk enzim *hs90a*

1. Output Ketepatan Klasifikasi Data *Training*

	pred.pos	pred.neg
real.pos	105	2

real.neg	14	304

2. Output Ketepatan Klasifikasi Data *Testing*

	pred.pos	pred.neg
real.pos	18	0
real.neg	1	56

LAMPIRAN E
Output R dari *Logistic Regression Ensemble* dengan Evaluasi *Cross Validation*

Treshold* Optimal pada enzim *aofb

Fold	1	2	3	4	5	6	7	8	9	10
Treshold	0.377483	0.375	0.369835	0.378926	0.373141	0.37562	0.373141	0.377273	0.376446	0.373141

Treshold* Optimal pada enzim *cah2

Fold	1	2	3	4	5	6	7	8	9	10
Treshold	0.373919	0.374751	0.37525	0.375749	0.376747	0.376081	0.373919	0.374751	0.37525	0.373586

Treshold* Optimal pada enzim *hs90a

Fold	1	2	3	4	5	6	7	8	9	10
Treshold	0.377778	0.371111	0.374444	0.377778	0.377778	0.372222	0.372222	0.374444	0.377778	0.374444

LAMPIRAN F

Program R untuk Pembagian Data *Training* dan Data *Testing*

```
enzim<-read.table("D:/enzim.txt",header=TRUE)  #load the data

#Split the data frame
splitDataFrame<-function(dataframe,seed=null,n=trainSize){
 if(!is.null(seed))set.seed(seed)
 index<-1:nrow(dataframe)
 trainindex<-sample(index,n)
 trainset<-dataframe[trainindex,]
 testset<-dataframe[-trainindex,]
 list(trainset=trainset,testset=testset)
 }
# Training Data 90% and Testing Data 10%
 split<-splitDataFrame(enzim,NULL,round(nrow(enzim)*0.90))
 train90<-split$trainset
 test10<-split$testset
write.csv(train90, "D:\\train90.csv")
write.csv(test10, "D:\\test10.csv")
```

LAMPIRAN G

Program R untuk *Logistic Regression Ensemble* (*Lorens*)

```
lr.cerp <- function(y,x,nens,fixsize=NULL,fixthres=NULL,search=F) {
        # initialization
        set.seed(as.numeric(Sys.time()))
        options(warn=-1)
        if(sum(is.na(x))>0) stop("missing value is found")
        if(sum(is.na(y))>0) stop("missing value is found")
        y <- as.data.frame(y)
        x <- as.data.frame(x)
        num_pred <- ncol(x)
        num_obs <- nrow(x)
        pos_rate <- sum(y)/num_obs

        # parameter search or default option
        if(search==T) {
                optimal <- search.thre_size(y,x,"lr")
                optsize <- optimal$size; opthreshold <-
optimal$threshold
        }
        else {
                if(is.null(fixsize)) fixsize <-
round(6*num_pred/num_obs)
                if(is.null(fixthres)) fixthres <- (pos_rate+.5)/2
                optsize <- fixsize; opthreshold <- fixthres
        }

        # main body
        ptss <- floor(seq(1,optsize+.999,length.out=num_pred))
        fitted <- NULL; predicted <- NULL; cname <- NULL;
coef.table<-matrix(0,num_pred,nens);
        partition.table <- matrix(0,num_pred,nens); intc <-
matrix(0,optsize,nens); probability <- rep(0,num_obs)
        for (i in 1:nens) {
                cname <- c(cname, paste("ens",i,sep=""))
                rand_pred <- sample(ptss)
                partition.table[,i] <- rand_pred
                avg_fit <- rep(0,num_obs)
```

```
                    for(j in 1:optsize) {
                            smp_dt <- cbind(y,x[,rand_pred==j])
                            intlr <-
glm(y~.,data=smp_dt,family=binomial())
                            coef.vector <- intlr$coefficient
                            coef.vector[is.na(coef.vector)] <- 0
                            intc[j,i] <- coef.vector[1]; coef.vector <-
coef.vector[-1]
                            coef.table[rand_pred==j,i] <- coef.vector
                            avg_fit <- avg_fit + intlr$fitted.values
                    }
                    fitted <- cbind(fitted,avg_fit/optsize,deparse.level=0)
                    probability <- probability+(avg_fit/optsize)/nens
            }
            learning.decision <- ens.voting(fitted,opthreshold)$final.vote
            colnames(fitted) <- cname
            colnames(intc) <- cname
            colnames(coef.table) <- cname; rownames(coef.table) <-
colnames(x)
            colnames(partition.table) <- cname; rownames(partition.table)
<- colnames(x)

            return(list(fitted=fitted,probability=probability,learning.decisio
n=learning.decision,

            partition.table=partition.table,coef.table=coef.table,intercept=in
tc,

            number.ensemble=nens,optimal.size=optsize,optimal.threshold
=opthreshold))
}

### lr.cerp.predict applies lr.cerp model to new data(test set) similar as
predict.lm function.
### lr.cerp.object is required and built from lr.cerp function.
### xtest is also required and should be same format as x in lr.cerp
function.
### ytest is optional if you want to check the accuracy
lr.cerp.predict <- function(lr.cerp.object,xtest,ytest=NULL) {
```

```
        # initialization
        options(warn=-1)
        if(sum(is.na(xtest))>0) stop("missing value is found")
        if(sum(is.na(ytest))>0) stop("missing value is found")
        xtest <- as.data.frame(xtest)
        num_obs <- nrow(xtest)
        nens <- lr.cerp.object$number.ensemble
        optsize <- lr.cerp.object$optimal.size
        opthreshold <- lr.cerp.object$optimal.threshold

        # main body
        cname <- NULL; test.decision <- NULL; fitted <- NULL;
probability <- rep(0,num_obs)
        xtest <- xtest[,rownames(lr.cerp.object$partition.table)]
        for (i in 1:nens) {
                avg_fit <- rep(0,num_obs)
                cname <- c(cname, paste("ens",i,sep=""))
                curmod <- lr.cerp.object$partition.table[,i]
                for(j in 1:optsize) {
                        intc <- lr.cerp.object$intercept[j,i]
                        wrkmat <- xtest[,curmod==j]
                        cvec <-
lr.cerp.object$coef.table[curmod==j,i]
                        int_vl <- as.matrix(wrkmat)%*%cvec
                        int_vl <- int_vl + intc
                        int_vl[int_vl>=709] <- 709
                        avg_fit <- avg_fit +
exp(int_vl)/(1+exp(int_vl))
                }
                fitted <- cbind(fitted,avg_fit/optsize,deparse.level=0)
                probability <- probability+(avg_fit/optsize)/nens
        }
        test.decision <- ens.voting(fitted,opthreshold,ytest)
        colnames(fitted) <- cname

        return(list(fitted=fitted,probability=t(probability),decision=test.
decision$final.vote,
```

```
        optimal.size=optsize,optimal.threshold=opthreshold,decision.ta
ble=test.decision$twobytwo))
}

### lr.cerp.cv performs v-fold cross-validation using lr.cerp and
lr.cerp.predict functions.
### Options and requirements are the same as lr.cerp function.
### One additional requirement is v_fold which is the number of fold to
be performed for cross-validation.
lr.cerp.cv <-
function(y,x,nens,v_fold,fixsize=NULL,fixthres=NULL,search=F) {
        # initialization
        set.seed(as.numeric(Sys.time()))
        options(warn=-1)
        if(sum(is.na(x))>0) stop("missing value is found")
        if(sum(is.na(y))>0) stop("missing value is found")
        y <- as.data.frame(y)
        x <- as.data.frame(x)
        num_obs <- nrow(y)
        rand_obs <- sample(1:num_obs)
        obs_rem <- num_obs%%v_fold
        obs_div <- (num_obs-obs_rem)/v_fold

        # main body
        probability <- rep(0,num_obs); predicted <- rep(0,num_obs);
tbttable <- matrix(0,2,2)
        part_size.list<-NULL; threshold.list<-NULL
        for(i in 1:v_fold) {
                if(i<=obs_rem) {head1<-(i-1)*(obs_div+1)+1;tail1<-
i*(obs_div+1);}
                else {head1<-(i-1)*obs_div+obs_rem+1;tail1<-
i*obs_div+obs_rem;}
                test_seq<-rand_obs[head1:tail1]
                learn_seq<-rand_obs[-c(head1:tail1)]
                ylearn<-y[learn_seq,];xlearn<-x[learn_seq,];xtest<-
x[test_seq,];ytest<-y[test_seq,]
                mid_rs<-
lr.cerp(ylearn,xlearn,nens,fixsize,fixthres,search)
```

```
                pred_rs<-lr.cerp.predict(mid_rs,xtest,ytest)
                predicted[test_seq]<-pred_rs$decision
                for(j in 1:nens) probability[test_seq]<-
probability[test_seq]+pred_rs$fitted[,j]/nens
                tbttable<-tbttable+pred_rs$decision.table
                part_size.list<-c(part_size.list,mid_rs$optimal.size)
                threshold.list<-
c(threshold.list,mid_rs$optimal.threshold)
        }

        return(
list(probability=probability,predicted=predicted,partition.size.list=part_s
ize.list,
                threshold.list=threshold.list,decision.table=tbttable))
}

### internal functions
ens.voting <- function (tot_res,threshold,y=NULL) {
        nens<-ncol(tot_res);nobs<-nrow(tot_res)
        if (!is.null(y)) {real_pos<-sum(y);real_neg<-nobs-real_pos}
        tot_res[tot_res>=threshold] <- 1; tot_res[tot_res<threshold] <-
0
        final.vote <- rep(0,nobs)
        for(i in 1:nobs) final.vote[i] <- mean(tot_res[i,])
        final.vote[final.vote>=0.5] <- 1; final.vote[final.vote<0.5] <- 0
        twobytwo <- NULL
        if (!is.null(y)) {
                real_pred_pos <- sum(final.vote==y&y==1)
                real_pred_neg <- sum(final.vote==y&y==0)
                real_pos_pred_neg <- real_pos - real_pred_pos
                real_neg_pred_pos <- real_neg - real_pred_neg
                twobytwo <-
rbind(c(real_pred_pos,real_pos_pred_neg),c(real_neg_pred_pos,real_pr
ed_neg))
                rownames(twobytwo) <- c("real.pos","real.neg")
                colnames(twobytwo) <- c("pred.pos","pred.neg")
        }
        return(list(final.vote=final.vote,twobytwo=twobytwo))
}
```

```
search.thre_size <- function (y,x,method) {
        nprd <- ncol(x);nobs <- nrow(x);orate <- sum(y)/nobs
        szseq <- NULL; int_fits <- NULL
        initseed <- c(2,3,4,5,6,7,8,9,10,12)
        for (i in initseed) {
                ipts<-i*nprd/nobs
                ipts<-floor(ipts)
                if (ipts%%2==0) ipts<-ipts+1
                if (szseq[length(szseq)]!=ipts||is.null(szseq)) {
                        szseq <- c(szseq,ipts)
                        int_fits <-
cbind(int_fits,cv.fit(y,x,ipts,method))
                }
        }
        nsrsz <- length(szseq)
        add_fits<-NULL;addsz<-NULL
        if(orate>=.5) iseq<-seq(.5,orate,.02)
        else {iseq<-seq(.5,orate,-.02); iseq<-rev(iseq)}
        nbis<-length(iseq)
        szfth<-rep(0,nbis);acfth<-rep(0,nbis)
        for(j in 1:nbis) {
                acseq<-rep(0,nsrsz)
                for(k in 1:nsrsz) {
                        tmpf<-rep(0,nobs)
                        tmpf[int_fits[,k]>=iseq[j]]<-
1;tmpf[int_fits[,k]<iseq[j]]<-0
                        acseq[k]<-sum(tmpf==y)/nobs
                }
                nbst<-sum(acseq==max(acseq));scol<-seq(1:nsrsz)
                if(nbst==1) nthc<-scol[acseq==max(acseq)]
                else {
                        tmpcol<-scol[acseq==max(acseq)]
                        nthc<-tmpcol[round(nbst/2)]
                }
                if(nthc==1) {
                        upts<-szseq[nthc+1];lpts<-szseq[nthc]
                        utfac<-acseq[nthc+1];ltfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
```

```
                                        mpts<-floor(mpts)
                                        if(mpts%%2==0) mpts<-mpts+1
                                        if(mpts==upts) break
                                        if(length(addsz)==0) {
                                                mtf<-
cv.fit(y,x,mpts,method)
                                                addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                        }
                                        else if(sum(addsz==mpts)==0) {
                                                mtf<-
cv.fit(y,x,mpts,method)
                                                addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                        }
                                        else mtf<-add_fits[,addsz==mpts]
                                        tmtf<-rep(0,nobs)
                                        tmtf[mtf>=iseq[j]]<-
1;tmtf[mtf<iseq[j]]<-0
                                        mtfac<-sum(tmpf==y)/nobs
                                        if(ltfac>utfac) {
                                                if(mtfac>=utfac) {upts<-
mpts;utfac<-mtfac}
                                                else {upts<-lpts;utfac<-
ltfac}
                                        }
                                        else if(ltfac<utfac) {
                                                if(mtfac>=ltfac) {lpts<-
mpts;ltfac<-mtfac}
                                                else {lpts<-upts;ltfac<-
utfac}
                                        }
                                        else {
                                                if(mtfac>=ltfac) {
                                                        lpts<-mpts;ltfac<-
mtfac
                                                        upts<-
mpts;utfac<-mtfac
                                                }
```

```
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                        }
                        if(ltfac>=utfac) {szfth[j]<-lpts;acfth[j]<-
ltfac}
                        else {szfth[j]<-upts;acfth[j]<-utfac}
                }
                else if(nthc==nsrsz) {
                        lpts<-szseq[nthc-1];upts<-szseq[nthc]
                        ltfac<-acseq[nthc-1];utfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
                                if(mpts==upts) break
                                if(length(addsz)==0) {
                                        mtf<-
cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else if(sum(addsz==mpts)==0) {
                                        mtf<-
cv.fit(y,x,mpts,method)
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else mtf<-add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-
1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
                                        if(mtfac>=utfac) {upts<-
mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
```

```
                                        else if(ltfac<utfac) {
                                                if(mtfac>=ltfac) {lpts<-
mpts;ltfac<-mtfac}
                                                else {lpts<-upts;ltfac<-
utfac}
                                        }
                                        else {
                                                if(mtfac>=ltfac) {
                                                        lpts<-mpts;ltfac<-
mtfac
                                                        upts<-
mpts;utfac<-mtfac
                                                }
                                                else {upts<-lpts;utfac<-
ltfac}
                                        }
                                }
                                if(ltfac>=utfac) {szfth[j]<-lpts;acfth[j]<-
ltfac}
                                else {szfth[j]<-upts;acfth[j]<-utfac}
                        }
                        else {
                                lpts<-szseq[nthc-1];upts<-szseq[nthc]
                                ltfac<-acseq[nthc-1];utfac<-acseq[nthc]
                                while(lpts!=upts) {
                                        mpts<-(lpts+upts)/2
                                        mpts<-floor(mpts)
                                        if(mpts%%2==0) mpts<-mpts+1
                                        if(mpts==upts) break
                                        if(length(addsz)==0) {
                                                mtf<-
cv.fit(y,x,mpts,method)
                                                addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                        }
                                        else if(sum(addsz==mpts)==0) {
                                                mtf<-
cv.fit(y,x,mpts,method)
```

```
                                        addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                }
                                else mtf<-add_fits[,addsz==mpts]
                                tmtf<-rep(0,nobs)
                                tmtf[mtf>=iseq[j]]<-
1;tmtf[mtf<iseq[j]]<-0
                                mtfac<-sum(tmpf==y)/nobs
                                if(ltfac>utfac) {
                                        if(mtfac>=utfac) {upts<-
mpts;utfac<-mtfac}
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                                else if(ltfac<utfac) {
                                        if(mtfac>=ltfac) {lpts<-
mpts;ltfac<-mtfac}
                                        else {lpts<-upts;ltfac<-
utfac}
                                }
                                else {
                                        if(mtfac>=ltfac) {
                                                lpts<-mpts;ltfac<-
mtfac
                                                upts<-
mpts;utfac<-mtfac
                                        }
                                        else {upts<-lpts;utfac<-
ltfac}
                                }
                        }
                        if(ltfac>=utfac) {lsps<-lpts;lsbs<-ltfac}
                        else {lsps<-upts;lsbs<-utfac}
                        upts<-szseq[nthc+1];lpts<-szseq[nthc]
                        utfac<-acseq[nthc+1];ltfac<-acseq[nthc]
                        while(lpts!=upts) {
                                mpts<-(lpts+upts)/2
                                mpts<-floor(mpts)
                                if(mpts%%2==0) mpts<-mpts+1
```

```
                                        if(mpts==upts) break
                                        if(length(addsz)==0) {
                                                mtf<-
cv.fit(y,x,mpts,method)
                                                addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                        }
                                        else if(sum(addsz==mpts)==0) {
                                                mtf<-
cv.fit(y,x,mpts,method)
                                                addsz<-
c(addsz,mpts);add_fits<-cbind(add_fits,mtf)
                                        }
                                        else mtf<-add_fits[,addsz==mpts]
                                        tmtf<-rep(0,nobs)
                                        tmtf[mtf>=iseq[j]]<-
1;tmtf[mtf<iseq[j]]<-0
                                        mtfac<-sum(tmpf==y)/nobs
                                        if(ltfac>utfac) {
                                                if(mtfac>=utfac) {upts<-
mpts;utfac<-mtfac}
                                                else {upts<-lpts;utfac<-
ltfac}
                                        }
                                        else if(ltfac<utfac) {
                                                if(mtfac>=ltfac) {lpts<-
mpts;ltfac<-mtfac}
                                                else {lpts<-upts;ltfac<-
utfac}
                                        }
                                        else {
                                                if(mtfac>=ltfac) {
                                                        lpts<-mpts;ltfac<-
mtfac
                                                        upts<-
mpts;utfac<-mtfac
                                                }
                                                else {upts<-lpts;utfac<-
ltfac}
```

```
                                        }
                                }
                                if(ltfac>=utfac) {usps<-lpts;usbs<-ltfac}
                                else {usps<-upts;usbs<-utfac}
                                if(lsbs>=usbs) {szfth[j]<-lsps;acfth[j]<-lsbs}
                                else {szfth[j]<-usps;acfth[j]<-usbs}
                        }
                }
                fnbst<-sum(max(acfth)==acfth);fscol<-seq(1:nbis)
                if(fnbst==1) {
                        finsz<-szfth[max(acfth)==acfth]
                        finth<-iseq[max(acfth)==acfth]
                }
                else {
                        ftmpcol<-fscol[max(acfth)==acfth]
                        tgcol<-ftmpcol[round(fnbst/2)]
                        finsz<-szfth[tgcol]
                        finth<-iseq[tgcol]
                }
                return(list(size=finsz,threshold=finth))
}

cv.fit <- function (y,x,npt,method) {
                num_pred<-ncol(x)
                num_obs<-nrow(x)
                lfit<-rep(0,num_obs)
                nv=3
                if(method=="lr") lfit<-lr.cerp.cv(y,x,1,nv)$probability
                else if(method=="lrt") lfit<-lrt.cerp.cv(y,x,1,nv)$probability
                else if(method=="ct") lfit<-ct.cerp.cv(y,x,1,nv)$probability
                return(lfit)
}
```

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS

Jainap Niken Melasasi, lahir di Trenggalek pada tanggal 20 Juni 1993, merupakan anak kedua dari tiga bersaudara. Penulis mulai menempuh jenjang pendidikan formal di bangku TK Jombang dan dilanjutkan hingga ke SDN Tunggorono 2 Jombang, SMPN 1 Jombang, SMAN 2 Jombang dan pada tahun 2011 terdaftar sebagai mahasiswa di Jurusan Statistika ITS melalui jalur masuk SNMPTN Undangan. Pengalaman perkuliahan selama 4 tahun, penulis mengisinya dengan mengikuti beberapa organisasi, kepanitian, kegiatan *training* dan kerja *part time*. Organisasi yang pernah saya ikuti adalah BEM FMIPA ITS sebagai staff perekonomian periode 2011/2012. Kepanitian yang pernah saya ikuti adalah panitia STATION (*Statistics Competition*), panitia FMIPA *Bisniss Plan* dan panitia Konferensi Nasional Matematika XVII. Selain kegiatan di kampus, penulis juga kerja *part time* di luar kampus yaitu sebagai guru les privat SD (semua mata pelajaran) dan sebagai Surveyor di sebuah *research marketing*. Penulis menerima segala saran, kritik, pertanyaan serta komentar yang membangun dari pembaca dapat melalui email atau No. Hp yaitu jainabniken@gmail.com atau 085745613700.